SOBERANÍA, RESISTENCIA Y AUTODEFENSA ELECTRÓNICA

EN TELÉFONOS MÓVILES:

PARTE 4

NUESTRAS COMUNICACIONES.
NUESTRA PRIVACIDAD.
NUESTRA LIBERTAD.

SIN VIGILANCIA.
SIN CENSURA.
SIN PERMISO.

CÓDIGO ABIERTO
MENTES LIBRES

SOFTWARE LIBRE
SOCIEDADES LIBRES

REDES LIBRES
VIGILANCIA CERO

COMPARTE. ESTUDIA.
MODIFICA. MEJORA.
HAZLO LIBRE.

EN ESTE DOCUMENTO:

- ANÁLISIS DE APLICACIONES DE MENSAJERÍA LIBRES Y SEGURAS: IRC, SIMPLEX, SESSION, XMPP, MATRIX, SIGNAL, ELEMENT, JAMI, TELEGRAM (ANÁLISIS CRÍTICO).
- PROTOCOLOS DE SEGURIDAD Y CIFRADO DE EXTREMO A EXTREMO.
- DESCENTRALIZACIÓN, FEDERACIÓN Y SOBERANÍA DIGITAL.
- PRIVACIDAD, METADATOS Y AUTODEFENSA FRENTE A LA VIGILANCIA.
- REQUISITOS TÉCNICOS, FORTALEZAS Y LIMITACIONES DE CADA PLATAFORMA.

PRIVACIDAD · AUTONOMÍA · RESISTENCIA · FEDERACIÓN · AUTODEFENSA · LIBERTAD

LA TECNOLOGÍA PUEDE SER UN ARMA DE CONTROL...
O UNA HERRAMIENTA DE LIBERACIÓN. TÚ ELIGES.

CRIPTOGRAFÍA PARA LA LIBERTAD

1. Aplicaciones de mensajería instantánea libres y seguras

Las aplicaciones de mensajería instantánea son herramientas clave para la comunicación digital. En el contexto de la soberanía y autodefensa electrónica, se valoran especialmente aquellas que son de código abierto, implementan cifrado de extremo a extremo y ofrecen modelos de comunicación descentralizados o federados. A continuación se analiza cada plataforma mencionada, destacando sus protocolos de seguridad, características de privacidad y arquitectura general.

1.1 IRC (Internet Relay Chat)

Internet Relay Chat es un protocolo de mensajería libre y descentralizado basado en texto. Funciona con arquitecturas cliente-servidor interconectadas: cualquier persona puede instalar un servidor IRC y conectarlo a una red global (por ejemplo EFNet, LiberaChat, Undernet, DALnet, TildeChat, AfterNET, QuakeNet o Rizon). Los clientes IRC envían mensajes en claro mediante TCP/Ports estándar, usualmente 6697 para texto cifrado y 6667 para texto no cifrado. Técnicamente, IRC usa un formato de comandos de texto plano (según RFC 1459 y posteriores) donde las comunicaciones van sin cifrar por defecto, aunque muchos servidores actuales aceptan conexiones TLS para encriptar el canal.

Su diseño federado (redes de servidores) otorga soberanía porque cualquiera puede montar su nodo sin permiso de un proveedor central. Sin embargo, sin capas extra de cifrado (como IRC sobre TLS o envoltorios de OTR) la privacidad es nula: cualquier observador en un punto de red o servidor puede leer direcciones (nicknames), canales y mensajes, violando confidencialidad e integridad de los datos. Los clientes modernos pueden complementar IRC con extensiones de cifrado como OTR (Off-the-Record) para chats uno a uno, añadiendo cifrado end-to-end usando dobles cifrados simétricos y DH efímero. A pesar de eso, IRC es vulnerable a metadatos expuestos (IP y nick visibles), extensiones limitadas de seguridad y censura fácil (un país podría bloquear puertos IRC o desactivar acceso a ciertos servidores). En consecuencia, se podría considerar que IRC ofrece libertad de despliegue y códigos abiertos, pero para ser seguro requiere adoptar voluntariamente TLS/OTR y confiar en redes seguras y descentralizadas.

- **Seguridad y cifrado:** Tradicionalmente, IRC no incluye cifrado por defecto: las conversaciones viajan en texto plano. Sin embargo, muchas implementaciones modernas permiten conectarse a los servidores IRC mediante TLS/SSL, lo que cifra la conexión entre cliente y servidor. Aun así, el contenido de los mensajes permanece sin cifrar a nivel de canal, a menos que los participantes usen herramientas adicionales (por ejemplo, cifrado

manual con PGP o plugins de cifrado punto a punto). En resumen, **IRC** ofrece cifrado de transporte (TLS) opcional, pero carece de cifrado integrado de extremo a extremo para los mensajes.

- **Privacidad:** Cada canal de IRC vive en un servidor, por lo cual los operadores del servidor pueden ver los usuarios conectados y el tráfico de mensajes. Al ser una arquitectura abierta, cualquiera puede montar un servidor IRC, pero eso implica confiar en el operador de dicho nodo. En términos de metadatos, se exponen la dirección IP de los usuarios al servidor IRC. Algunos clientes permiten funcionalidades extra como conectarse vía Tor o VPN para ocultar la IP.
- **Descentralización:** La red IRC es federada/descentralizada en el sentido de que existen múltiples servidores independientes que pueden interconectarse. Sin embargo, carece de un estándar unificado para esta interconexión (cada red utiliza su propio protocolo de enlace o gateways). Es más descentralizado que un servicio cerrado, pero la falta de cifrado es una debilidad frente a la vigilancia.
- **Requisitos técnicos y disponibilidad:** Hay múltiples clientes IRC para dispositivos móviles (por ejemplo, AndroIRC, Quassel, IRCCloud) y para escritorio. Puede funcionar en prácticamente cualquier sistema operativo. No requiere número de teléfono, solo un apodo (nickname) y servidor.
- **Fortalezas:** Simplicidad, ligereza en recursos, amplio soporte. Puede ser útil como canal de chat abierto en entornos donde no se exija alta privacidad.
- **Limitaciones:** Ausencia de cifrado E2E, exposición de metadatos, moderación localizada por operadores de canales. No es óptimo para comunicaciones sensibles sin herramientas adicionales.

1.2 Simplex

SimpleX es una red de mensajería descentralizada diseñada para máxima privacidad sin que existan identificadores fijos. No usa números telefónicos ni cuentas registradas: cada usuario dispone de direcciones anónimas temporales.

La arquitectura de SimpleX consiste en una red de múltiples servidores independientes a los que el usuario decide conectarse; cada chat utiliza por lo menos cuatro nodos distintos, dispersando los mensajes para impedir correlación de IPs. Emplea cifrado de extremo a extremo de alta seguridad: todos los mensajes se protegen con claves asimétricas y simétricas que se renuevan continuamente, incluyendo intercambio post-cuántico de claves (por ejemplo integrando algoritmos basados en curvas elípticas junto con encapsulación ML-

KEM tipo NIST). La información de perfil, contactos y metadatos no se guarda en ningún servidor: solo el cliente almacena la base de datos cifrada localmente.

A diferencia de la mensajería común, SimpleX no asigna IDs estáticos sino "direcciones anónimas puntuales" (cadenas generadas al vuelo para cada contacto). Esto evita espionaje de metadatos: los servidores nunca saben quién se comunica con quién. Los servidores solo guardan mensajes cifrados temporalmente mientras el receptor está desconectado, luego los eliminan.

Como contramedidas, SimpleX permite usar Tor para ocultar la IP del cliente. Su protocolo (SMP y Chat Relay) fue desarrollado de forma abierta, aunque es menos conocido que otros. En práctica, SimpleX destaca por su fuerte modelo de control de datos: objetos críticos como claves y mensajes son exclusivos del cliente, obteniendo plena soberanía digital.

No obstante, al depender de usuarios que hospeden servidores, su seguridad final radica en la confianza del administrador del servidor (aunque tampoco ve el contenido en claro). En resumen, SimpleX combina cifrado extremo a extremo (incluyendo futuras protecciones post-cuánticas) con ausencia de metadatos fijos, ofreciendo una arquitectura verdaderamente descentralizada que fortalece la autonomía comunicativa.

- **Protocolos de seguridad:** Integra el **Signal Protocol** para cifrado de extremo a extremo en chats individuales y grupales. Al estar construido sobre tecnologías como XMPP o Matrix, aprovecha protocolos descentralizados, pero agrega su propia capa de cifrado fuerte basada en **OMEMO** y el modelo de ratchet doble (Double Ratchet).
- **Privacidad:** Al ser open source, cualquier persona puede auditar el código. Se promueve la privacidad evitando la vinculación al número telefónico y permitiendo crear cuentas con direcciones de email o identificadores propios. Al utilizar servidores de XMPP o Matrix, dependen de la política de cada servidor en cuanto a retención de datos. Simplex parece ofrecer también opciones para cifrar archivos y mensajes multimedia.
- **Descentralización y federación:** Simplex aprovecha la red XMPP federada o las matrices descentralizadas, lo que significa que no hay un único punto de falla. Cada usuario puede elegir o montar su propio servidor. El cifrado E2E asegura que ni el servidor admin tiene acceso al contenido.
- **Requisitos técnicos:** Disponible para dispositivos móviles (Android e iOS) y plataformas de escritorio. Requiere compatibilidad con los protocolos XMPP o Matrix. Dado que es relativamente nueva, su adopción es limitada comparada con apps más establecidas.

- **Fortalezas:** Seguridad sólida gracias a Signal Protocol y cifrado E2E, descentralización inherente por el uso de XMPP/Matrix, multiplataforma, open source.
- **Limitaciones:** Al depender de otras redes (XMPP/Matrix), comparte sus retos: necesidad de registrar una cuenta en algún servidor, potencial fragmentación de servidores, usuarios limitados (puede ser difícil encontrar contactos fuera de plataformas comunes). La calidad de la experiencia puede variar según el servidor elegido.

1.3 Jabber/XMPP

XMPP (Extensible Messaging and Presence Protocol), antes llamado Jabber, es un protocolo abierto y federado para mensajería. Permite que cualquier persona corra su propio servidor (por ejemplo ejabberd, Prosody) y se conecte libremente con otros servidores. La arquitectura es plenamente federada: los datos de un usuario están controlados por el dueño de su servidor (o "pod"), quien se encarga de identidad, copia de seguridad, políticas. Esto otorga soberanía digital ya que no hay empresa central: un usuario puede moverse de servidor o montar el suyo sin perder la capacidad de comunicarse. Originalmente XMPP no cifra mensajes; sin embargo, existen varias capas de seguridad que se agregan como XEPs. En primer lugar, el canal cliente-servidor puede asegurarse con TLS, protegiendo la transferencia básica.

Para la seguridad end-to-end, XMPP adopta extensiones populares como OMEMO (XEP-0384) o OTR (Off-the-Record). OMEMO implementa un protocolo basado en Double Ratchet (compatible con Signal Protocol), que permite cifrar mensajes de grupo o individuales con forward secrecy y ratchet de clave. También se pueden usar PGP para cifrar mensajes en XMPP, aunque OMEMO permite intercambios multi-dispositivo. Además, XMPP soporta SRV records y DNSSEC para autenticar servidores.

En cuanto a metadatos, un servidor XMPP sabe quién envía mensajes a quién (característica intrínseca de la federación), aunque no puede leer su contenido cifrado. Se han propuesto (pero no implantado ampliamente) mecanismos para ofuscar metadatos como OFT o redes de mezcla. En general, las ventajas de XMPP son su flexibilidad (personalización por XEPs) y comunidad abierta. Por su federación, es resistente a la censura institucional: bloquear un servidor implica migrar a otro. Sin embargo, cada servidor impone su propio nivel de privacidad, y la federación puede ser fragmentada si algunos servidores no permiten cifrado o filtran usuarios. En resumen, XMPP brinda una base técnica sólida y modular (usando XML stanzas), extensiones de cifrado modernas y la independencia de

servidores, lo cual beneficia la autonomía comunicativa, siempre y cuando los usuarios activen OMEMO/OTR y elijan proveedores confiables.

- **Protocolos de seguridad:** Originalmente no cifraba contenido de extremo a extremo. Sin embargo, se puede usar **OMEMO** (Multi-End Message and Object Encryption) para proveer cifrado E2E. OMEMO implementa el Signal Protocol adaptado para XMPP, con encriptación ratcheada y soporte para chats grupales. También se puede usar **OTR (Off-the-Record Messaging)** para cifrado puntual en chats uno a uno. La conexión del cliente al servidor se cifra típicamente con TLS (porque XMPP suele requerir una conexión cifrada moderna).
- **Privacidad:** XMPP no requiere vinculación a un teléfono o identidad real; cada usuario tiene un JID (ej. usuario@servidor.org). La privacidad depende del servidor: el contenido se guarda en el servidor hasta su entrega (a menos que se use cifrado E2E, en cuyo caso el servidor solo ve datos cifrados). Los metadatos (direcciones, tiempos de conexión) pueden ser conocidos por el servidor. Al ser federado, cada servidor administra sus propios logs y políticas.
- **Descentralización y federación:** XMPP es altamente descentralizado: cualquier persona puede montar un servidor Jabber, y todos los servidores pueden federarse (conectarse) entre sí. De esta forma, un usuario en `servidorA.com` puede hablar con un usuario en `servidorB.net`. Esta federación permite la libre interconexión sin confiar en una única entidad. Además, XMPP es extensible vía XEPs (XMPP Extension Protocols) que agregan funcionalidades (chat en grupo, archivos, video, etc.).
- **Requisitos técnicos:** Hay numerosos clientes XMPP para móvil y escritorio (Conversations, Xabber, BeagleIM, Gajim, etc.). Soporta Android, iOS, Windows, Linux. Requiere que el usuario escoja o cree un servidor Jabber; muchos servicios ofrecen cuentas gratis (como cuenta en `jabber.org`, `dismail.de`, etc.).
- **Fortalezas:** Comunidad madura y amplia compatibilidad; open source; fácil de auditar. Federación realza la soberanía, pues cualquiera decide en qué servidor confiar. Con OMEMO, ofrece confidencialidad de contenidos de manera robusta.
- **Limitaciones:** Configurar usuarios nuevos puede ser menos intuitivo que apps que requieren solo un teléfono. La dependencia de los servidores implica que si un servidor es comprometido, podría ver los metadatos o intentar ataques activos, aunque OMEMO mitiga mucho eso. La

experiencia de usuarios avanzados es excelente, pero algunos usuarios casuales pueden considerarlo complejo.

1.4 Session

Session es un mensajero enfocado en meta-privacidad y descentralización total. Funciona sin números telefónicos ni cuentas centrales. Su arquitectura combina una red de mensajería P2P basada en una capa de onion routing anónima con sincronización por blockchain ligera. En concreto, Session usa la base técnica de Signal Protocol (AES-256 para mensajes, X25519 para intercambio de claves, ratchet DH), pero sustituye sus servidores centrales por una red descentralizada de Servidores Session y utiliza direcciones basadas en blockchain Oxen para identificar nodos, lo que impide metadatos de usuarios. Cada usuario obtiene una clave pública larga ("Session ID") que equivale a su dirección. Para enviar un mensaje, el cliente busca a través de la red Lokinet (una red anónima similar a Tor) al destinatario y enruta el mensaje sin revelar IP ni servidor fijo. Session implementa proxies y reenvíos intermedios (mezclas) para ocultar correspondencia de mensajes.

Técnicamente integra cifrado end-to-end (ratchet de doble/single DH plus AES) igual que Signal, pero todo ocurre a través de nodos onion; además utiliza un esquema de reenvíos sincronizado de mensajes (no hay push central). No mantiene metadatos, pues no hay servidor con registros de quien se comunica con quién. A nivel de seguridad, Session soporta características como mensajes efímeros y no retiene chats en servidores tras la entrega. La descentralización (red de nodos voluntarios) fortalece la resistencia a la censura: no existe punto central que se pueda cerrar o forzar legalmente. Como contrapartida, la complejidad de la red onion puede incrementar latencia y depende de nodos confiables para enrutado. En definitiva, Session fusiona el sólido cifrado de Signal con infraestructura descentralizada onion + blockchain, proporcionando mensajería metadato-rresistente y anónima, ideal para la soberanía porque retira la dependencia de cualquier servicio único o proveedor de identidad.

- **Protocolos de seguridad:** Session usa un fork del Signal Protocol adaptado a su arquitectura descentralizada. Cada usuario recibe un par de llaves criptográficas y un identificador anónimo, sin revelar información de identidad. El cifrado es de extremo a extremo: cada mensaje se cifra con la llave pública del destinatario y se mezcla en la red distribuida de servidores para dificultar el rastreo de metadatos.
- **Privacidad:** Session sobresale en la reducción de metadatos: no requiere ningún dato personal para registrarse, ni siquiera direcciones IP (ya que el

tráfico viaja por la red de privacidad Oxen, similar a Tor pero basada en Proof-of-Stake y mix-nodos). Esto significa que teóricamente ni los operadores de nodos pueden relacionar usuarios con mensajes. Además, incorporan *"off-the-record groups"*, donde no queda historial en ningún servidor.

- **Descentralización:** A diferencia de Signal (que tiene servidores centrales para gestión de datos criptográficos), Session se apoya en una red distribuida de nodos (nodos Lokinet/Oxen), evitando cualquier servidor central. Este enfoque blockchainmente distribuido permite autopublicación y resiliencia: si se desconecta un nodo, otros mantienen la red viva. La federación al estilo XMPP no aplica, pues Session es su propio ecosistema, pero es completamente P2P-descentralizado.
- **Requisitos técnicos:** Aplicación disponible para Android, iOS y escritorio. No necesita cuenta; simplemente genera claves al instalarla. Opera lento en comparación con mensajeros convencionales debido a su diseño anónimo y relés entre nodos, pero la compensación es mucha más privacidad.
- **Fortalezas:** Privacidad extrema, casi total anonimato. Sin identificación, sin metadatos visibles. End-to-end encryption garantizada. No depende de entidades centralizadas. Codebase abierto y auditado parcialmente.
- **Limitaciones:** La experiencia de usuario puede ser más lenta (mensajes tardan más en llegar) debido a la red mix de nodos. La adopción aún es pequeña, por lo que encontrar contactos en la app puede ser difícil. Carece de algunas funciones avanzadas (como llamadas de voz/video) presentes en otras apps. Al estar en una red blockchain, hay riesgo teórico de ataques relacionados con la cadena de bloques, aunque su diseño impide la censura.

1.5 Signal

Signal se consolida como el estándar de oro en mensajería segura, ofreciendo una arquitectura técnica robusta y 100% de código abierto (licencia AGPLv3) que prioriza la privacidad por diseño sobre la conveniencia comercial. Disponible para Android, su implementación permite compilaciones reproducibles que verifican la integridad criptográfica del binario; sin embargo, su compatibilidad directa con el repositorio oficial de F-Droid es compleja debido a dependencias de servicios propietarios y requisitos estrictos de firma, lo que lleva a la organización a recomendar la descarga directa desde signal.org o el uso de forks adaptados (como Molly) para entornos estrictamente libres. En su núcleo, el Protocolo Signal combina el acuerdo de claves X3DH (Extended Triple Diffie-

Hellman) con el algoritmo Double Ratchet, garantizando secreto hacia adelante y hacia el futuro: incluso si una clave de sesión es comprometida, los mensajes pasados y futuros permanecen indescifrables.

Además, funciones avanzadas como "Sealed Sender" (Remitente Sellado) cifran los metadatos del remitente, minimizando drásticamente la huella digital y evitando que el servidor central sepa quién envía el mensaje, una protección crítica para activistas, periodistas y profesionales en escenarios de alto riesgo. A diferencia de otras plataformas, Signal no almacena metadatos de comunicación (quién, cuándo o desde dónde), no incluye rastreadores, anuncios ni minería de datos, y mantiene la telemetría reducida al mínimo absoluto, todo respaldado por una estructura de organización sin fines de lucro independiente de grandes corporaciones tecnológicas.

Cada mensaje, llamada de voz o video y archivo transferido está protegido por defecto con cifrado de extremo a extremo (E2EE), sin modos "opcionales" de privacidad ni puertas traseras, lo que permite auditorías de seguridad independientes y continuas. Esta combinación de rigor criptográfico, transparencia en el desarrollo y compromiso con la soberanía digital convierte a Signal en una herramienta indispensable tanto para la resistencia digital colectiva y grupos sin fines de lucro, como para la protección cotidiana de datos personales, asegurando que la confidencialidad y la integridad de la comunicación se mantengan intactas frente a la vigilancia masiva y los ataques sofisticados.

- **Protocolos de seguridad:** Utiliza la Signal Protocol, un esquema criptográfico moderno que combina criptografía de curva elíptica (X25519), autenticación de clave pública (Ed25519) y un algoritmo ratchet doble para generar llaves efímeras. Esto permite **perfect forward secrecy** (cada mensaje tiene llave única, con derivación continua), e **inmunidad a compromisos pasados**. El protocolo cifra mensajes de texto, archivos, llamadas y videollamadas. También usa *sealed sender* (remitente sellado) para ocultar la identidad del emisor incluso al servidor de Signal.
- **Privacidad:** Signal almacena mínima metadata. No retiene conversaciones en servidores (solo colas temporales para entrega); no guarda direcciones de contacto ni almacena registros de quién envía mensajes a quién (excepto por una marca de tiempo general cuando se unió). Todos los datos críticos están encriptados en el dispositivo. Requiere número telefónico para registro (y opcionalmente PIN), lo cual es su principal limitación en

términos de anonimato: la cuenta está vinculada al número, aunque no accede a la lista de contactos del teléfono en sus servidores.

- **Descentralización:** Signal no es federado; opera en una infraestructura de servidores centralizados de la fundación Signal. Esto facilita la gestión de usuarios y la confianza de que el código en servidores sea auditado, pero implica confiar en una organización. Aun así, con cifrado E2E, Signal no puede leer los mensajes; la centralización solo afecta la disponibilidad del servicio (requiere que sus servidores funcionen).
- **Requisitos técnicos:** App disponible en Android, iOS, Windows, macOS, Linux. Las conversaciones se sincronizan entre dispositivos usando la cuenta telefónica y PIN (los usuarios pueden vincular desktops). Para usar en PC, el teléfono debe tener la app activa. Soporta grupos cifrados, mensajes temporales, notas en tiempo real, llamadas seguras.
- **Fortalezas:** Seguridad robusta, auditada, y fácil de usar. Popularidad amplia (muchos usuarios ya lo usan), lo que facilita encontrar contactos. Funciones completas (chats grupales, chats temporales, multimedia). Transparencia de código abierto. Protección contra captura de pantalla en chats secretos (opcional).
- **Limitaciones:** Requiere número de teléfono (no anónimo). El servidor centralizado es un punto de falla (aunque sin acceso a contenido). Algunos consideran que la dependencia del fundación sin respaldo descentralizado compromete la *soberanía total* digital. Además, si el protocolo central fuera bloqueado por operadores (como ha sucedido en algunos países), los usuarios lo percibirían como servicio inaccesible.

1.6 Matrix

Matrix es un protocolo abierto de mensajería y VoIP federado, diseñado para interoperabilidad y cifrado nativo. Cada usuario crea cuenta en un homeserver (público o propio), que gestiona salas y almacena mensajes. Los servidores Matrix se federan mediante un esquema de replicación: cuando A y B hablan, ambos servidores intercambian eventos JSON a través de HTTP push. Matrix utiliza cifrado de extremo a extremo llamado Olm/Megolm. Olm se emplea para el establecimiento de sesiones (basado en Double Ratchet de axolotl), mientras Megolm cifra mensajes de grupo con ratchets simétricos avanzados para permitir suscripción de nuevos dispositivos.

Toda la comunicación cifrada (texto, medios) se distribuye de modo que, en salas cifradas, solo los usuarios con claves pueden leerlos, y los servidores registran únicamente datos encriptados. La arquitectura federada de Matrix permite que

organizaciones o individuos puedan ejecutar su servidor (o emplear servicios federados) y seguir comunicándose con otros. Esto fortalece la soberanía porque no hay control único: si un nodo es atacado o bloqueado, se puede migrar. Matrix también soporta puentes con otras redes (como Slack, IRC, Telegram), pero lo crítico es que es «open-source everything».

La descentralización implica a menudo mayor tolerancia a la censura: países pueden bloquear instancias pero seguir usándola si los usuarios hallan instancias externas o VPN. Como limitantes, Matrix requiere configuración precisa (certificados, congestión federación) y, en chats grandes, Megolm debe regenerar llaves para prevenir filtraciones post-compromiso. En síntesis, Matrix provee un protocolo robusto basado en HTTP/XML/JSON, cifrado potente (Ratchet avanzado), replicación de datos entre servidores federados, y enfoque libre, lo cual otorga control real a los usuarios sobre dónde se aloja su información.

- **Protocolos de seguridad:** Para cifrado de extremo a extremo, Matrix usa **Olm**/**Megolm**, que implementan un sistema similar al ratchet doble de Signal (Olm) para conversaciones 1:1 y un esquema de cifrado grupal (Megolm). Soporta también el estándar de comunicación segura *Matrix E2E*, con autenticación mutua de llaves. Además, las conexiones de cliente a servidor se cifran con TLS/HTTPS, protegiendo las credenciales y transporte de mensajes.
- **Privacidad:** Cada servidor guarda el historial de mensajes y datos de los usuarios alojados en él. Los chats cifrados, sin embargo, son almacenados cifrados. Los metadatos como quién pertenece a qué sala (room) pueden ser visibles para el servidor, pero el contenido real está cifrado. Al ser federado, los usuarios pueden elegir su servidor (o instalar el propio), controlando dónde se aloja su información. Dado que cada mensaje cifrado solo puede ser descifrado con la llave del destinatario (o llaves del grupo), los servidores federados no pueden leer los mensajes privados.
- **Descentralización:** Matrix es intrínsecamente federado. Cualquier entidad puede ejecutar un servidor Matrix (homeserver). Existen proveedores populares (modular.im, etc.), pero también instalaciones privadas. Todos los servidores federados comparten un "log de eventos", así cualquier sala pública puede ser accesible desde cualquier servidor. Esto permite interoperar como una red global, donde un usuario de un servidor habla con otro sin intermediarios centralizados.
- **Requisitos técnicos:** Clientes disponibles para Android, iOS, Windows, macOS, Linux, e incluso web (por ejemplo, Element, FluffyChat). Requiere nombre de usuario@servidor. El registro puede ser abierto (en algunos

servidores) o controlado. Las llamadas de voz/video requieren un bridge con WebRTC, y la calidad depende del servidor y configuración.

- **Fortalezas:** Gran compatibilidad (chats, voz, video, archivos), fuerte cifrado habilitado, autonomía del usuario (puede autohospedarse), comunidad activa desarrollando funciones. Buen rendimiento en salas grupales, con historial accesible.
- **Limitaciones:** El cifrado grupal Megolm es más ligero criptográficamente que Ratchet de Signal; hay un riesgo de que, si comprometen la llave de sesión del grupo, puedan leer todo el chat grupal. La federación implica a veces poca garantía de que un servidor ajeno sea seguro. Como proyecto en rápido crecimiento, pueden existir bugs y no todas las funciones están maduras (por ejemplo, migración de salas, limitaciones de ancho de banda). Para un usuario no técnico, elegir un servidor federado puede ser confuso al inicio.

1.7 Briar

Briar es un sistema de mensajería diseñado para entornos hostiles y desconectados. Funciona peer-to-peer sin servidores, aprovechando Bluetooth/Wi-Fi locales o la red Tor cuando hay Internet. Cada usuario mantiene una base de datos cifrada local; los mensajes se envían directamente entre dispositivos cercanos o, si no hay radio, se encolan en una red segura vía Tor a través de nodos intermedios.

Lo esencial de Briar es su resistencia a la censura: al no depender de ninguna infraestructura central o proveedores de Internet, un régimen represivo no puede bloquear un servidor porque no existe. La seguridad se basa en claves públicas de tamaño corto (Briar ID) que se intercambian solo cuando dos dispositivos físicamente cercanos (Bluetooth) sincronizan sus claves manualmente, evitando cualquier rastreo remoto. Cada mensaje usa cifrado end-to-end (generalmente AES-256 y técnicas de la librería Curve25519 para intercambio de claves).

Adicionalmente, Briar anonimiza el meta-tráfico: las solicitudes sobre la red Tor (para enviar/recibir) están cifradas dentro de múltiples capas onion, de modo que ninguno de los puntos intermedios conoce el origen final del mensaje. La arquitectura no tiene servidores, de modo que los datos nunca se exponen a terceros; los mensajes se entregan directos del emisor al receptor, o se almacenan cifrados temporalmente en nodos seleccionados con la llave pública del destinatario, y luego se auto-eliminan.

Las ventajas soberanas incluyen la autosuficiencia completa (solo necesitas una app instalada), anonimato total del paradero y minimización de metadatos (no hay listas globales de contacto; sólo existen en cada dispositivo). Los vectores de ataque son limitados a ataques físicos o malware local; un adversario no puede interceptar remoto sin acceso directo al dispositivo. En resumen, Briar enfatiza la autodefensa electrónica: opera en un entorno offline resistente, con cifrado fuerte y red Tor, garantizando confidencialidad y disponibilidad incluso bajo vigilancia extrema.

- **Protocolos de seguridad:** Utiliza cifrado asimétrico: cada usuario tiene un par de llaves criptográficas (identidad pública/privada). Las conversaciones se cifran con **AES de 256 bits** y autenticadas con RSA/4096. Los mensajes no salen del dispositivo hasta estar cifrados. Las claves se intercambian por proximidad (escaneando códigos QR) para evitar suplantación.
- **Privacidad:** No requiere servidor externo; los mensajes viajan directamente de dispositivo a dispositivo (capa local) o a través de Tor. Así se evita la existencia de un servidor que almacene historiales o metadatos. Las comunicaciones se anclan en la aplicación de cada usuario: una copia de cada conversación solo está en los dispositivos de los participantes, nunca en servidores. Por tanto, no hay almacén central de datos personales ni de mensajes.
- **Descentralización:** Completamente descentralizado. En zonas sin conectividad a internet (por ejemplo, áreas rurales o entornos opresivos), puede usarse en modo offline (Mesh network), donde los dispositivos se conectan en red local temporal. Cuando se restablece conexión a internet, se enrutan los mensajes cifrados por la red Tor automáticamente.
- **Requisitos técnicos:** Aplicación para Android y próximamente en otros sistemas (hasta ahora, foco Android). No necesita registro con ningún dato personal, solo crea la identidad interna en la app. Se puede enlazar con otras cuentas (por ejemplo, foros o servicios) sin comprometer la privacidad.
- **Fortalezas:** Privacidad sobresaliente: ni servidores ni empresas conocen los mensajes. Funciona sin internet (ideal para escenarios de censura o de campo). Protege contra vigilancia gracias a Tor y a la ausencia de backends.
- **Limitaciones:** Experiencia de usuario centrada en chat básico (no tiene llamadas integradas). Depende de proximidad inicial para activar la comunicación. La escalabilidad para grandes grupos puede ser limitada debido a la infraestructura peer-to-peer. Actualmente, solo Android es oficialmente soportado (aunque hay esfuerzos de sincronización con la PC).

1.8 Tox

TOX es una red de mensajería P2P descentralizada, sin servidores centrales. Cada usuario genera su propio ID (una clave pública larga) y los contactos se agregan mediante intercambio de estas IDs. TOX emplea una arquitectura tipo DHT (Distributed Hash Table) para el descubrimiento de pares: los usuarios se comunican sobre una red de nodos voluntarios que almacenan información de contacto cifrada.

En el plano de seguridad, todos los mensajes y llamadas son cifrados con algoritmos estándar (por ejemplo, criptografía NaCl: X25519 para intercambio de claves, Salsa20+Poly1305 o ChaCha20-Poly1305 para cifrado simétrico). La autenticación peer-to-peer significa que no existe autoridad central con la que registrarse. Esto brinda soberanía pues no hay un intermediario que pueda revocar cuentas ni sus metadatos (sólo los usuarios conocen su malla de contactos).

TOX soporta mensajería de texto, voz, video y transferencia de archivos, todo cifrado end-to-end. Al ser puro P2P, es resistente a la censura convencional; sin embargo, requiere puertos abiertos o NAT traversal (usando UDP hole punching o Tor) para conectar usuarios, lo que en redes muy controladas puede bloquearse. Un inconveniente es que la calidad de la conexión depende directamente de los pares (puede haber retardos si el enrutado DHT es lento).

En cuanto a privacidad, TOX no es totalmente inmune al análisis de tráfico: un observador global podría inferir patrones de comunicación analizando direcciones IP de los nodos que se conectan, a menos de usar Tor adicionalmente. Aun así, su propuesta de valor es clara: mensajes cifrados punto a punto sin depender de empresas ni metadatos centrados, otorgando mayor control personal sobre los datos.

- **Protocolos de seguridad:** Usa **NaCl/libsodium** para cifrado. Cada usuario genera una clave pública y privada al crear su identidad Tox. Las comunicaciones (texto, audio, video, archivos) se cifran con estas llaves usando tecnologías modernas (Curve25519 y XSalsa20-Poly1305). Tox garantiza el cifrado end-to-end en todas sus formas de mensajería.
- **Privacidad:** Al ser P2P puro, no hay servidores que puedan almacenar o espiar mensajes. Los contactos se agregan mediante un identificador único Tox ID, que no revela información personal. La dirección IP de los usuarios es visible para sus contactos directos (pues se conectan uno a uno), pero esto se oculta parcialmente mediante un sistema de *DHT (Distributed Hash*

Table) que enruta las conexiones sin necesidad de direcciones IP directas, actuando como un overlay distribuido.

- **Descentralización:** Completamente P2P. Tox no tiene nodos centrales que puedan fallar; en su lugar, los clientes forman una red donde cada nodo ayuda al enrutamiento de los demás. Cualquiera puede hospedar un "bootstrap" o nodo de apoyo, pero no es esencial para el funcionamiento entre pares existentes. Esta estructura de red hace a Tox muy resistente a la censura estatal (no hay organización a apagar) pero confiable en redes abiertas.
- **Requisitos técnicos:** Existen múltiples clientes basados en la especificación Tox: qTox (Windows/Linux), Antox (Android), uTox (multiplataforma ligera), Ricin (multiplataforma), entre otros. Funciona en dispositivos de bajo consumo si se requiere. El emparejamiento se hace compartiendo la Tox ID del usuario.
- **Fortalezas:** Seguridad y privacidad integrales sin configuraciones complicadas. Total independencia de empresas. Además de mensajería, provee llamadas de voz y video cifradas. Fácil de auditar por ser de código abierto.
- **Limitaciones:** Al ser red P2P, el rendimiento depende de conexiones de los pares: llamadas de voz/video pueden tener latencia o desconexiones si alguna parte tiene problemas. Añadir contactos requiere intercambio manual de ID (lo que dificulta onboarding masivo). La interfaz de algunos clientes puede ser rudimentaria comparada con apps comerciales.

1.9 Olvid

Olvid es una aplicación enfocada en entornos empresariales de alta seguridad. Su diseño diferencial es que cifra no sólo los mensajes sino también los metadatos. Es un servicio centralizado en cuanto a la existencia de un servidor (similar a Slack corporativo), pero su arquitectura criptográfica descansa exclusivamente en claves de usuario, de modo que los servidores nunca ven datos útiles en claro. Técnicamente, Olvid utiliza criptografía personalizada: cada mensaje tiene su propia clave de cifrado efímera y las conversaciones se basan en flujos criptográficos certificados entre clientes. Olvid garantiza la autenticidad de interlocutores mediante un sistema de credenciales fuertemente asegurado, posiblemente empleando certificados X.509 o un esquema equivalente (aunque no hay TTP central).

Toda la información de usuario (nombres, fotos, directorio) va cifrada de extremo a extremo. Los servidores de Olvid solamente canalizan paquetes cifrados, sin

almacenarlos (salvo metadata mínima de entrega temporal). Según sus diseñadores, implementa firme Forward Secrecy y revocación segura de claves: si una clave maestra se ve comprometida, pasados los chats quedan aún protegidos por claves efímeras no almacenadas. Además, recibió certificaciones de ANSSI (agencia francesa de seguridad) como prueba de su implementación robusta. En la práctica, Olvid se distingue porque incluso elementos como IDs de mensaje o relaciones entre ellos quedan ofuscados: sin clave, nadie puede reconstruir quién se habló con quién.

Su enfoque privado es compatible con entornos corporativos donde se necesita cero filtración de metadata. La contrapartida es que depende de servidores (técnicamente "confiables" de forma que no retengan nada; pero en un modelo cerrado, si el servidor se cae la conversación cesa). Sin embargo, en cuanto a soberanía digital, Olvid demuestra que es posible crear un chat centralizado con garantías criptográficas totales: el proveedor del servidor (empresa que lo despliega) carece de la capacidad de escuchar o analizar comunicación alguna. Esto lo hace ideal para escenarios donde se combina la conveniencia empresarial con exigencias de confidencialidad estricta.

- **Protocolos de seguridad:** Usa criptografía de curva elíptica (Elliptic Curve Cryptography) para cifrar mensajes punto a punto. Cada mensaje se cifra con una clave única generada aleatoriamente (clave simétrica efímera) y la llave pública del destinatario, asegurando que ni siquiera los propios servidores conozcan las llaves. Implementa un mecanismo de *CHAUM*, donde cada mensaje enviado genera una clave que luego se destruye tras leerlo. Ello garantiza que el mensaje dejado en el dispositivo destino sea la única copia encrypte-fácil.
- **Privacidad:** Olvid no mantiene servidores intermediarios que almacenen permanentemente los mensajes; en su lugar, utiliza servidores *efímeros* que retienen el texto cifrado únicamente hasta que este es entregado a su destinatario. Una vez entregado, el contenido se borra definitivamente del servidor, reduciendo la superficie de espionaje. El servicio exige identificación mediante dirección de correo o cuenta existente, pero no obliga al uso de un número telefónico. Incluye características de mensajes autodestructibles y rastreo de lectura.
- **Descentralización:** No es totalmente descentralizado en el sentido de P2P o federación abierta: Olvid gestiona su propia red de servidores temporales. Sin embargo, su diseño busca que los servidores no tengan conocimiento de las claves de encriptación. En este sentido, ofrece

"autonomía" en la custodia de datos (ni siquiera los administradores pueden leerlos).

- **Requisitos técnicos:** Disponible para Android e iOS. Requiere registro con email. Permite chats individuales y grupales. Se integra con Active Directory en entornos corporativos (ocupa nicho en empresas que buscan mensajería interna segura).
- **Fortalezas:** Cifrado sólido, ethos de eliminación de datos, no dependen de terceros para lectura. Buena solución para mensajería confidencial.
- **Limitaciones:** No es software libre (aunque su arquitectura de cifrado es transparentada, el código no es totalmente abierto). Su enfoque corporativo puede limitar algunas funciones de flexibilidad social. Al no ser ampliamente popular, tiene ecosistema limitado de usuarios.

1.10 Vector

Vector es un mensajero cifrado libre que se apoya en la red Nostr (Notes and Other Stuff Transmitted by Relays) para la infraestructura de transporte. Nostr es un protocolo abierto para mensajería que usa relays descentralizados. Cada usuario tiene un par de claves (pública/privada) y se suscribe a eventos de relays.

Técnicamente, Vector implementa estándares Nostr de cifrado (por ejemplo NIP-44, NIP-59 y especialmente NIP-17 para mensajería privada). Estos NIPs combinan técnicas de cifrado asimétrico (curve25519) y enveloping ("gift wrap") para garantizar que solo el destinatario puede desencriptar.

Al utilizar Nostr, Vector no guarda mensajes en servidores propios: cualquier relay (servidor público) sirve como pasarela de datos, pero sólo almacena textos cifrados, no metadatos asociados a identidad real. La ventaja principal es ausencia de KYC y anonimato ante los desarrolladores: dado que los usuarios solo intercambian sus claves Nostr, Vector no puede relacionar las cuentas con identidades reales.

En la práctica, Vector también soporta chat de texto y voz cifrada, y llamadas P2P obligando a los nodos relay solo actuar como intermediarios ciegos. Su código abierto y la elección de relays federados lo alinean con la soberanía: el usuario puede cambiar de relay si detecta censura, o correr el suyo. Al usar Nostr, Vector hereda protección de acceso (requiere firma de mensajes con claves) y se beneficia de la sencillez del protocolo (publicaciones JSON).

Sin embargo, Nostr es un protocolo relativamente nuevo, así que su nivel de madurez y auditoría es menor. Básicamente, Vector apuesta por una capa de abstracción novedosa para la privacidad: ausencia de meta-direcciones (solo claves aleatorias), cifrado end-to-end vía Nostr standards, descentralización de los servicios de mensajería (relays). Todo ello crea un entorno resistente a la vigilancia centralizada, fortaleciendo la autonomía y la resistencia a bloqueos a nivel de servidor.

- **Protocolos de seguridad:** Aparentemente implementa cifrado de extremo a extremo usando librerías basadas en Signal Protocol. Es de código abierto bajo la organización VectorPrivacy, por lo que el código criptográfico puede revisarse. Las conversaciones individuales y grupales se cifran en el dispositivo con llaves del par usuario (similares a Signal o Olvid).
- **Privacidad:** No requiere número de teléfono; en cambio usa nombre de usuario y PIN. No retiene datos de usuarios ni contenidos en servidores, de acuerdo a su filosofía. Se enfoca en minimizar metadatos. Cada mensaje temporal solo queda en la memoria del servidor hasta su entrega y después se borra.
- **Descentralización:** Aunque mantiene servidores propios para la gestión de mensajes (permite alta disponibilidad), su diseño propone una arquitectura federada basada en blockchain (Vector premisa que fusiona mensajería con blockchain para evitar censura). En práctica, esto significa que las transacciones de mensajes podrían replicarse en una cadena de bloques privada para auditoría, pero los contenidos siguen cifrados E2E.
- **Requisitos técnicos:** Disponibles apps para Android, iOS y web. Requiere registro inicial con email/nick. Dado que es reciente, está en desarrollo, con la idea de integrar más funciones próximamente.
- **Fortalezas:** Código abierto, cifrado E2E, sin requerir teléfono, enfoque en privacidad.
- **Limitaciones:** Proyecto emergente, poca adopción. Detalles exactos de su uso de blockchain y federación pueden ser complejos de entender para el usuario final. Al no tener un ecosistema amplio, dificulta que se conecte con otros contactos fuera de sus servidores.

1.11 Wire

Wire es una plataforma de mensajería cifrada de origen suizo, enfocada en empresas y grupos colaborativos. Técnicamente, está construida con tecnología similar a Signal: utiliza el protocolo Proteus, diseñado por Wire sobre la base del Signal Protocol (doble ratchet de X25519/DH para forward secrecy). Cada chat,

archivo o llamada se encripta de extremo a extremo con AES-256-GCM. Para llamadas de voz y video, Wire emplea DTLS y SRTP (protocolos estándar de cifrado en tiempo real).

La arquitectura habitual de Wire es cliente-servidor: los servidores centrales de Wire gestionan intercambios de mensajes cifrados y colas de entrega. Sin embargo, los datos de usuario no se almacenan en los servidores más allá del tiempo necesario.

Wire es de código abierto (código cliente GPL) y ofrece opciones de despliegue on-premise, lo que favorece la soberanía para organizaciones (pueden hospedar sus propios servidores Wire). El cifrado E2E previene que ni siquiera Wire como empresa vea el contenido de las conversaciones.

Adicionalmente, Wire permite la creación de cuentas anónimas (sin número telefónico) para usuarios finales, e integra características de autenticación (p.ej. inicio de sesión por correo electrónico con contraseñas derivadas con scrypt). En seguridad, Wire ha sido auditado y tiene cierta certificación alemana BSI (autoridad de seguridad).

El control de metadatos es mejor que en mensajeros comerciales: Wire minimiza registros del uso (propagando información solo por una hora, cero logging de mensajes). Sin embargo, la naturaleza cliente-servidor implica cierto riesgo si los servidores fueran comprometidos: aunque no tienen las claves privadas de cifrado, podrían revelar activos de metadatos (número de dispositivos, actividad de backup cifrada).

En síntesis, Wire ofrece cifrado robusto estilo Signal, arquitectura descentralizada posible para grupos y un diseño enfocado en negocios, brindando soberanía en el sentido de autogestión de infraestructura y fuerte cifrado; sigue siendo menos totalmente descentralizado que soluciones P2P, pero mucho más seguro que mensajería masiva convencional.

- **Protocolos de seguridad:** Usa un cifrado E2E robusto basado en **Proteus/MQTT** (un protocolo basado en Axolotl/Double Ratchet), similar al Signal Protocol. Cada mensaje se cifra con claves únicas de curva elíptica, y Wire emplea también autenticación de llaves públicas. Además, las llamadas de voz y video están cifradas con **ZRTP** (pionero en cifrado de llamadas).
- **Privacidad:** Wire es de código abierto (excepto algunos componentes de audio/video), y afirma no retener contenidos ni registros de contactos. Los servidores de Wire solo manejan mensajes cifrados, sin llaves para

descifrarlos. Exige registro por correo o cuenta de empresa, pero no vinculada obligatoriamente a un teléfono. Además, Wire aplica estándares de privacidad europeos (respetando GDPR), y se autoaudita con firmas externas.

- **Descentralización:** A diferencia de XMPP o Matrix, Wire no es federado. Sin embargo, permite múltiples despliegues: las empresas pueden auto-hospedar un servidor Wire para su organización. Para usuarios individuales, se usan los servidores en la nube de Wire (filiales en Suiza). Desde el punto de vista de usuario final, la mensajería ocurre a través de servidores Wire, pero con cifrado extremo a extremo, lo que significa que la empresa Wire no ve el contenido.
- **Requisitos técnicos:** Aplicaciones para Android, iOS, Windows, macOS, Linux. Soporta cuentas personales y de equipo (indicado para entornos colaborativos). Wire permite chats privados y grupales, llamadas de voz, videollamadas, notas de audio y compartir archivos. También provee "coberturas" temporales, mensajes con expiración, gestión de dispositivos autorizados.
- **Fortalezas:** Cifrado fuerte, enfoque corporativo lo hace atractivo para empresas (por su cumplimiento legal), multiplataforma. Buen desempeño en llamadas.
- **Limitaciones:** Al no ser federado, confía en que Wire permanezca un servicio saludable. Aunque de código abierto, la parte de voz/video no lo es completamente. Requiere email o cuenta y aceptan el modelo de negocio. A nivel de usuario, es similar a usar cualquier mensajería centralizada (solo que los admin no pueden leer contenido).

1.12 Zerion

Zerion es una aplicación de mensajería orientada a máxima privacidad y resistencia futura. Opera exclusivamente sobre la red Tor, lo que significa que nunca expone la IP real de los usuarios ni utiliza ningún servidor central común. La comunicación es estrictamente peer-to-peer; solo fluye entre contactos directos, usando Tor hidden services para cada usuario. Cada usuario tiene un identificador sin relación con identidad real. Zerion incorpora una criptografía de vanguardia: implementa una Triple Ratchet con claves post-cuánticas. En cada mensaje se aplica una "polarización híbrida": combina X25519 (crypto clásica) con un esquema PQ (por ejemplo, Mode1-Full: Kyber-768 encapsulado en cada mensaje).

Esto asegura que incluso si un adversario guarda mensajes para descifrarlos en el futuro con un computador cuántico, el cifrado actual de Zerion ya es resistente. La aplicación cifra no solo mensajes sino también contactos y chats completos en una “bóveda cifrada” local en el dispositivo. Además, Zerion añade medidas anti-forense (higiene de logs, eliminación segura, contraseñas de hardware). En términos de arquitectura, Zerion transmite todo solo a través de Tor, así que los datos viajan encapsulados en Tor v3; esto proporciona una fuerte, sólida y robusta capa de anonimato.

Adicionalmente, cuenta con funciones avanzadas como canales unidireccionales (difusión) y verificación de identidad entre contactos. Los chats individuales son end-to-end cifrados con AES-256-Poly1305 y claves efímeras; los grupos usan una clave de grupo rota frecuentemente. Como ventaja soberana, Zerion no depende de ningún dominio ni infraestructura de terceros: incluso los desarrolladores o autoridades no pueden vincular la información de los usuarios, ni los mensajes tienen metadatos visibles (solo existe la topología Tor habitual).

Su gran ventaja es el uso de Tor (aunque necesita buena conectividad) y un target algo “exótico” (poco usado fuera de círculos muy técnicos). Sin embargo, Zerion demuestra que es posible elevar la seguridad de mensajería al máximo nivel criptográfico, blindando la privacidad personal contra la vigilancia futura y dando al usuario total control de su clave maestra y datos.

- **Protocolos de seguridad:** Usa **XMPP con OMEMO** para cifrado E2E. Esto implica que las conversaciones entre usuarios Zerion están cifradas con llaves de curva elíptica, manteniendo perfect forward secrecy. Aprovecha los estándares establecidos en XMPP para garantizar el cifrado sin depender de un servidor central malicioso.
- **Privacidad:** Siguiendo la filosofía XMPP, cada mensaje cifrado solo puede ser descifrado por el destinatario. Zerion evita recopilar datos de usuarios, y como es open source, cualquiera puede inspeccionar el código. No requiere número telefónico; los usuarios se identifican con un nombre de usuario en el dominio de su servidor. Los mensajes se encolan en los servidores solo si el receptor no está en línea, pero permanecen cifrados hasta su lectura.
- **Descentralización:** Zerion propone una red federada de servidores basados en XMPP. Cualquier administrador puede correr su propia instancia de servidor Zerion y conectar con otras instancias. De este modo, se promueve la soberanía digital: comunidades u organizaciones pueden

operar su propio servidor para sus miembros, sin depender de entidades comerciales.

- **Requisitos técnicos:** Existe aplicación móvil (probablemente Android e iOS) así como clientes de escritorio. Al estar enfocado en XMPP, puede interoperar con otros clientes OMEMO compatibles. Es multiplataforma en la medida que se utilice XMPP.
- **Fortalezas:** Sigue un enfoque probado (XMPP+OMEMO). Promueve la federación e interoperabilidad. Transparencia de código.
- **Limitaciones:** Zerion aún es un proyecto pequeño; su ecosistema es emergente. Para usuarios, no es plug-and-play ya que precisa conocer el modelo XMPP. Si bien es libre, la adopción general es baja, lo que limita la red de contactos.

1.13 ArcaneChat

ArcaneChat es un cliente de mensajería móvil diseñado como extensión de DeltaChat, con características de privacidad reforzadas. Funciona sobre la infraestructura de correo electrónico (como DeltaChat), pero agrega capas de anonimato y cifrado automático. Cada usuario crea cuentas “fantasmas” (bindings de correo sin datos reales), y la comunicación se cifra con PGP (sistema Autocrypt) de extremo a extremo: los mensajes viajan como emails encriptados adjuntos, invisibles al servidor SMTP/IMAP.

Al basarse en correo, ArcaneChat hereda la federación global del email: cualquier proveedor de email puede actuar como “servidor de chat”. Esto brinda soberanía tecnológica porque no hay una sola compañía dueña de la red, y los usuarios pueden elegir servidores de correo nacionales o privados según su confianza. ArcaneChat simplifica la gestión de claves (las intercambia en segundo plano) y soporta múltiples identidades en un mismo dispositivo (perfiles independientes).

En cuanto a protocolos, usa IMAP/SMTP estándar pero con contenidos encriptados, y posibilita conversaciones 1:1 y grupos (mediante listas de distribución cifradas). Desde el punto de vista de seguridad, el cifrado PGP garantiza que ni el proveedor de correo ni intermediarios lean el contenido. Los metadatos de header del email sí quedan (remitente/destinatario cifrados con OpenPGP/MIME), pero el asunto y cuerpo quedan ocultos.

ArcaneChat mejora la privacidad añadiendo autenticación anónima: no requiere teléfono ni datos personales, solo un alias de email. Sus desarrolladores enfatizan que el servidor de ArcaneChat (que maneja sincronización entre dispositivos) nunca retiene mensajes más de 20 días, y elimina datos al cierre de sesión. De

esta forma, se puede entender que ArcaneChat es, en esencia, DeltaChat con interfaz móvil mejorada: aprovecha la descentralización e inercia del correo (casi imposible de censurar a gran escala) y cifrado forte, facilitando el autocontrol sobre la plataforma de mensajería.

1.14 Cwtch

Desarrollada por Open Privacy Research Society, Cwtch es una aplicación de mensajería descentralizada, de código abierto y multigrupo, diseñada específicamente para ofrecer una resistencia avanzada a metadatos y una privacidad robusta en Android.

Su arquitectura híbrida combina comunicación peer-to-peer (P2P) para chats individuales y un modelo cliente-servidor para grupos, pero con una distinción crítica: los servidores son completamente no confiables (*untrusted*), lo que impide que la infraestructura de retransmisión acceda a cualquier dato sensible.

Técnicamente, el núcleo de Cwtch está implementado en Go (con bindings en Rust y Dart/Flutter para la interfaz de usuario en Android), garantizando que toda la comunicación esté cifrada de extremo a extremo y se enruute exclusivamente a través de servicios onion Tor v3, ocultando eficazmente las direcciones IP y neutralizando el análisis de tráfico de red.

A diferencia de las plataformas convencionales, Cwtch minimiza deliberadamente la filtración de canales laterales y metadatos, asegurando que ni siquiera la existencia de una conexión o la topología de la red sea visible sin consentimiento explícito.

En el dispositivo Android, todos los perfiles, claves criptográficas y mensajes se almacenan localmente y cifrados en el disco, eliminando por completo la telemetría, la dependencia de ecosistemas propietarios y el riesgo de filtraciones en la nube. Esta soberanía de datos permite a los usuarios (desde activistas en entornos de alta vigilancia hasta profesionales y uso cotidiano) crear o unirse a "espacios seguros" descentralizados, donde la federación es opcional y la infraestructura puede ser autoalojada o comunitaria.

En su documentación se refleja un compromiso con la transparencia y la seguridad verificable, ofreciendo herramientas para compilaciones reproducibles en Android, lo que permite a organizaciones sin fines de lucro y grupos de resistencia digital auditar el código, verificar la ausencia de puertas traseras y mantener la independencia tecnológica frente a la vigilancia masiva.

- **Protocolos de seguridad:** Cwtch se basa en **Sphinx** y **Ratcheting**. Los mensajes se enrutan a través de una red de nodos (Sphinx, similar a las mezclas de Tor) de forma anónima, haciendo difícil rastrear emisores. Cada chat genera una llave pública efímera, y se utilizan propuestas similares a `Ratcheting.NET` para mantener forward secrecy. Todo el contenido va cifrado de extremo a extremo con curvas elípticas.
- **Privacidad:** No requiere identificación externa; se generan direcciones anónimas. Los usuarios configuran conocidos introduciendo las claves unos de otros localmente (por ejemplo, escaneando códigos QR o enviando clave via medios seguros). Dado que los mensajes viajan por la red Cwtch anónima, ni los nodos ni el backend (si lo hubiera) conocen quién envía mensajes a quién. Incluso en caso de toma del servidor, no hay datos legibles.
- **Descentralización:** Usa la red Tor (u Onion Services) para enrutar datos. Cada usuario se conecta mediante un "cliente" que crea hidden services en Tor, permitiendo que sus contactos envíen mensajes mediante esas direcciones .onion. No depende de servidores centrales; cada usuario actúa como servicio oculto Tor. Esto asegura que la comunicación sea doblemente protegida: anonimato de Tor + cifrado E2E.
- **Requisitos técnicos:** Disponible para Windows, Linux, macOS y Android (en desarrollo). No requiere cuenta tradicional; el usuario configura un nombre interno. Se enfoca en chats individuales, aunque se planea soporte para grupos mediante soluciones criptográficas.
- **Fortalezas:** Privacidad excepcional: anonimato de red y cifrado. Muy resistente a espionaje de ISP o gobierno. Código abierto por la organización Open Privacy (la misma detrás de GNUnet, Cryptoparty, etc.).
- **Limitaciones:** Más complejo de usar: requiere instalación local, se recomienda mínimo conocimiento de redes. Rendimiento dependiente de Tor (más latencia). Aún en fase temprana de desarrollo, con funciones limitadas frente a mensajería convencional (por ejemplo, sin videollamadas integradas). El uso de Tor puede impedir su uso en redes muy restringidas o detectables por DPI.

1.15 DeltaChat

DeltaChat es un mensajero descentralizado que utiliza el sistema de correo electrónico como capa de transporte. Cada usuario se identifica con una dirección de email normal; la aplicación se conecta a servidores de correo (IMAP/SMTP) configurados por el usuario. En el nivel técnico, DeltaChat envía mensajes como

correos electrónicos cifrados con OpenPGP. Cuando un contacto responde, llega como correo en la bandeja de entrada del destinatario, y el cliente DeltaChat lo presenta como chat. Esta arquitectura federada (cualquier servidor de correo puede participar) evita dependencia de un proveedor central y aprovecha la infraestructura global existente.

Desde la perspectiva de la soberanía, usar email significa que no es necesario confiar en un servicio nuevo o bloqueable: uno puede usar cualquier servicio de correos cifrado o propio. La seguridad se basa en la criptografía OpenPGP: al crear un chat, los usuarios intercambian claves públicamente a través de mensajes previos. Cada mensaje de chat viaja cifrado end-to-end, y la metadata no cifrada se limita a los encabezados mínimos (que no revelan el contenido ni interlocutores, pues éstos también se codifican).

DeltaChat respeta la confidencialidad, ya que sólo los dispositivos con la clave privada del usuario pueden descifrar el contenido de mensajes. Como método de resistencia a la vigilancia, es notable que bloquear el chat requeriría bloquear el acceso al correo electrónico en general, lo cual es mucho más difícil (salvo cortar todos los servicios de correo en un país).

Sin embargo, la calidad de servicio depende de la latencia del email (los mensajes pasan por servidores tradicionales) y de la fuerza de las contraseñas, además de la confianza en los clientes de correo usados. En práctica, DeltaChat es ideal para proporcionar mensajería cifrada sobre una plataforma ubicua, promoviendo la autodeterminación tecnológica al basarse en protocolos abiertos bien establecidos (SMTP/IMAP y OpenPGP).

- **Protocolos de seguridad:** Implementa **Autocrypt** y **PGP** para cifrado de extremo a extremo. Al enviar un mensaje, DeltaChat cifra el contenido con la llave pública del destinatario (usando PGP) y lo envía como un correo electrónico. Si el destinatario también tiene DeltaChat, lo reconoce y lo trata como un mensaje de chat en la app. Para proteger metadata, las comunicaciones PGP/E2E implican que los correos son legibles solo para los usuarios, aunque las cabeceras (direcciones de email) siguen visibles. DeltaChat cifra el cuerpo y archivos multimedia. Además, los mensajes en la app se manejan en chats encadenados, similares a mensajería convencional.
- **Privacidad:** Utilizar el correo como medio significa que cualquier persona con una dirección de email (y usando DeltaChat) se convierte instantáneamente en contacto, sin necesidad de cuentas adicionales. No hay servidores nuevos más allá del servidor de email propio (IMAP/SMTP).

Los datos de conversación se alojan en las cuentas de email de los usuarios (según configuración IMAP); así, puede decirse que los usuarios tienen control total sobre sus mensajes (están en su cuenta de correo, que pueden auto-hospedar o elegir un proveedor fiable).

- **Descentralización:** La infraestructura es la existente de internet: cualquier servidor de correo (Gmail, ProtonMail, servidor privado) puede servir a DeltaChat sin modificaciones. Esto hace al sistema globalmente interoperable. No hay un servidor central de DeltaChat: la aplicación funciona con las cuentas de cada usuario. Por lo tanto, la descentralización está garantizada a nivel de correo.
- **Requisitos técnicos:** Apps para Android, iOS, Windows, macOS, Linux. El usuario debe tener o crear una cuenta de correo (IMAP/SMTP). La propia DeltaChat puede configurar proveedores comunes automáticamente (Gmail, Outlook, etc.) si el usuario provee credenciales. No necesita teléfono ni registro extra.
- **Fortalezas:** Aprovecha infraestructura ya existente (muy resistente a bloqueos específicos). Fácil de usar: si conoce usar email, puede chatear. Cifrado E2E con PGP (si se configura Autocrypt o intercambia llaves). Libertad de elegir proveedor de correo o servidor.
- **Limitaciones:** Si el proveedor de correo no permite PGP, podría requerir configuración manual. La experiencia depende de la sincronización de correo (puede haber retrasos si el servidor tarda en entregar). Los metadatos de email (remitente, asuntos) quedan expuestos, a menos de ocultar asunt a cosas triviales. No tiene características propias de mensajería (por ej., reacciones rápidas, stickers, etc.). Para emergencias, si se corta el acceso al correo, se pierde la mensajería (aunque ese es un problema de la infraestructura de email, no de DeltaChat misma).

1.16 Armada

Armada es una aplicación de comunicación comunitaria de código abierto (licencia AGPL-3.0), disponible para Android a través de Zapstore, diseñada sobre los protocolos Nostr y Concord. Su arquitectura representa un paradigma avanzado en autodefensa digital, ofreciendo una alternativa descentralizada y cifrada de extremo a extremo (E2EE) a las plataformas de chat corporativas tradicionales.

En el ámbito de la privacidad y la seguridad, Armada implementa el protocolo Concord, que garantiza que todos los mensajes, metadatos, identificadores de comunidad y nombres de canales estén sellados criptográficamente en el

dispositivo del usuario antes de su transmisión. Mediante el uso de cifrado de envoltorio (NIP-59), los relés de la red solo procesan datos ilegibles, imposibilitando que terceros, incluidos los proveedores de infraestructura, intercepten o analicen el contenido. Además, las llamadas de voz utilizan un sistema de tokenización ciega (CORD-07) que preserva el anonimato de los participantes.

Respecto a la soberanía digital, la aplicación elimina la dependencia de identificadores centralizados como números de teléfono o correos electrónicos. La identidad se gestiona exclusivamente mediante claves criptográficas portátiles (nsec, NIP-07 o NIP-46), otorgando al individuo la propiedad absoluta de su presencia digital y facilitando la migración entre dispositivos sin intermediarios. Esto neutraliza riesgos de vigilancia masiva y expulsión arbitraria, ya que no existe una entidad corporativa con capacidad para eliminar cuentas o disolver comunidades de manera global.

La resiliencia operativa se logra mediante un modelo "serverless" por defecto. Las comunidades se distribuyen a través de múltiples relés Nostr independientes. Si un nodo es censurado o sufre una interrupción, la red se adapta y la comunicación continúa ininterrumpidamente en los nodos restantes, garantizando la resistencia ante la censura. Para organizaciones o usuarios que requieran control total, Armada admite el autoalojamiento (*self-hosting*) mediante relés compatibles con NIP-29, permitiendo la gestión autónoma de la membresía, la moderación y el almacenamiento de datos.

No obstante, la autodefensa digital exige rigor continuo. Se recomienda verificar los permisos solicitados por el paquete de Android (evitando privilegios innecesarios como `android.permission.DUMP`, reportados en versiones específicas) y validar la integridad de las compilaciones. En conjunto, Armada constituye una herramienta robusta para la acción colectiva, el periodismo independiente y la preservación de datos personales, alineándose con los más altos estándares de cifrado, federación y soberanía tecnológica.

- **Protocolos de seguridad:** Nostr, en esencia, no encripta los mensajes: las publicaciones y eventos enviados a los relays son legibles para cualquiera que la lea. Sin embargo, se basa en criptografía de firma ECDSA para asegurar la autenticidad de los mensajes. Armada, como cliente de Nostr especializado en chat, puede ofrecer capa de cifrado E2E sobre Nostr usando un esquema extra (por ejemplo, intercambiando llaves de Curve25519 entre usuarios). No hay datos oficiales detallados, pero

presumiblemente sigue métodos similares a ArcaneChat para cifrar mensajes directos.

- **Privacidad:** Al igual que ArcaneChat, Nostr no requiere credenciales tradicionales: los usuarios se identifican por su clave pública (o por un apodo inversible a esa clave). Los mensajes enviados a relays están públicamente disponibles (puede ser un desafío para privacidad de broadcast público), pero en chats directos cifrados (cuando Armada los implementa), sólo los participantes pueden leerlos. No implica servidores privados, sino relays comunitarios que retienen eventos.
- **Descentralización:** Total. Nostr se define como «de relays»: una red de mensajes en la que cualquier persona puede correr un relay (servidor) y cualquier cliente puede leer/escribir en ellos. Armada es uno de varios clientes Nostr (junto con damus, iris, etc.). No hay autoridad central. Esto la hace altamente resistente a la censura, aunque los relays pueden filtrar qué mensajes propagan (pero normalmente comprometer un número grande de relays es complicado).
- **Requisitos técnicos:** Se accede por cliente de Android (Armada) u otros clientes. Solo se necesita generar un par de llaves en la app. Usa Internet normal.
- **Fortalezas:** Libertad total de red; sin limitaciones geográficas impuestas por un servidor grande. Como parte de Nostr, fomenta la comunicación como publicación-noticia, que puede ser útil para difundir información.
- **Limitaciones:** Todavía incubado; no hay muchas guías de uso. Mensajes públicos en relays no son confidenciales. Depende de desarrollos futuros para estandarizar chats cifrados privados. La comunidad es muy pequeña comparada con mensajerías establecidas. En resumen, es prometedor pero muy nuevo.

2. Aplicaciones de redes sociales libres y seguras

Las redes sociales federadas y descentralizadas ofrecen espacios para compartir contenido multimedia y mensajes de forma más soberana que las plataformas centralizadas. Abogan por la privatización del control de datos y la resistencia a la censura. Las plataformas listadas siguen estándares abiertos (sobre todo ActivityPub) o modelos similares que permiten la interacción entre redes distintas. A continuación, se explora cada una.

2.1 Primal

Primal es una aplicación de red social descentralizada construida sobre el protocolo Nostr (Not Only Social, Temporally Repliable). Nostr es un protocolo simple de publicación/following que no requiere servidores centrales. Cada usuario de Primal tiene claves públicas que representan su identidad y publica eventos (como «toots») en relays (servidores públicos o comunitarios). Los relays simplemente difunden los mensajes cifrados con la clave pública del suscriptor. Primal añade a Nostr un componente de cartera bitcoin integrada y enfoque en «medios sociales descentralizados».

Al usar Nostr, Primal no sufre censura de un proveedor único: los contenidos se replican en múltiples relays redundantes, que cualquier persona puede montar. Los usuarios eligen a quién seguir publicando su clave, garantizando que no hay autorización central para participar. En cuanto a privacidad, Nostr (y Primal) suelen ser públicos: las publicaciones se pueden cifrar sólo si el autor lo desea con la clave del receptor. Sin embargo, no hay datos de usuarios recopilados por la plataforma central; la única información compartida es la necesaria para publicar/ver mensajes (firmados con la clave privada del autor). Esto da cierto nivel de soberanía, pues los metadatos globales son mínimos y el código es abierto.

A pesar de ello, la censura técnica es difícil a nivel de relays: un país puede cerrar un relay pero no toda la red. Primal, en particular, se promociona como una interfaz moderna sobre Nostr, posiblemente con énfasis en imágenes y una cartera (se rumora integración Lightning). Su modelo descentralizado y «sin custodia» permite a los usuarios retener el control total de sus datos. Cabe mencionar que Primal al estar en desarrollo activo (parte de Web3/Bitcoin stacks) representa un cambio de paradigma: usuarios entran usando solo claves, sin registrar nada personal. Esto maximiza la resistencia a la vigilancia, aunque el proyecto es nuevo y el ecosistema de relays aún se consolida.

2.2 Pixelfed

Pixelfed es una red social federada especializada en intercambio de imágenes (similar a Instagram, pero descentralizado). Su protocolo base es ActivityPub (misma especificación que Mastodon). Cada comunidad Pixelfed (pod) actúa como servidor independiente al que los usuarios se suscriben para compartir fotos.

Los pods se federan naturalmente: usuarios de un pod pueden seguir y comentar las fotos de usuarios de otros pods. Técnicamente, cuando un usuario sube una

foto, ésta se almacena en su servidor local; luego su servidor envía notificaciones (“activities”) al servidor de cada seguidor, replicando metadatos (título, descripciones) y proporcionando enlaces al fichero. Para seguridad, Pixelfed implementa cifrado HTTPS para transferencia cliente-servidor.

Sin embargo, ActivityPub no cifra por defecto contenido, así que si un pod decide no cifrar backend, los administradores podrían ver los medios. En la práctica muchos pods respetan privacidad (no analizan datos del usuario). La soberanía digital de Pixelfed radica en el federado: cualquiera puede auto-hospedar un pod e imponer sus propias políticas.

No hay censura obligatoria de un ente global; la moderación se realiza pod a pod. Además, la privacidad es mayor que en servicios centralizados: uno puede elegir pods bajo jurisdicciones con leyes de datos favorables o pod conocidos por no compartir info.

Debido a su foco en imágenes, Pixelfed incentiva menos recolección de metadata personal (en comparación con redes de texto), pero permite hashtags y seguidores públicos. Al ser libre y usar ActivityPub, Pixelfed respeta estándares abiertos y facilita interoperar con otras plataformas del Fediverso (por ejemplo, puede compartir contenidos con Mastodon o Friendica).

En suma, Pixelfed demuestra que una red social de multimedia puede ser distribuida: aunque un nodo puede recoger cierta info, el usuario decide su servidor y, en principio, no está obligado a exponer sus imágenes a vigilancia de un gran monopolio. Esto contribuye a la soberanía digital al permitir a la comunidad mantener y moderar su red de imágenes sin intermediarios comerciales.

- **Arquitectura y soberanía de datos:** Cada usuario crea una cuenta en un servidor (pod) Pixelfed. Los servidores pueden ser auto-hospedados o públicos. Los usuarios pueden cargar fotos y videos, los cuales se almacenan en el servidor donde se publicó la cuenta del autor. Cada servidor es independiente, y el usuario puede elegir cuál le inspira confianza, teniendo control sobre la moderación y la retención de datos de su contenido.
- **Federación:** Los servidores Pixelfed federan con la red ActivityPub. Esto significa que un usuario en un servidor Pixelfed puede seguir y comentar contenido de usuarios en otros servidores Pixelfed, Mastodon, Pleroma, Friendica y demás compatibles con ActivityPub. Esta interoperabilidad amplía mucho las conexiones posibles.

- **Privacidad:** Las cuentas en Pixelfed pueden ser públicas o privadas. Los contenidos privados solo se muestran a seguidores aprobados. Los servidores no comparten meta información sensible (solo los posts necesarios para federar). Al ser software libre, cada pod determina sus propios términos: algunos retienen logs por un tiempo, otros los limpian regularmente.
- **Seguridad y cifrado:** El tráfico al servidor Pixelfed se debe cifrar con HTTPS (como cualquier web moderna). Sin embargo, las imágenes en sí no van cifradas de extremo a extremo por defecto; una vez publicadas en un servidor público, cualquiera federado las recibe en claro. Para contenido sensible, Pixelfed permite cuentas privadas con aprobación manual de seguidores. No hay cifrado E2E nativo en Pixelfed (no es mensajería), pero el acceso al servidor está autenticado por credenciales.
- **Datos y políticas:** Al ser software libre, el código es abierto, pero los datos quedan en manos de los servidores. No existe seguimiento publicitario interno (salvo los pods particulares que podrían exponer contenido con técnicas de SEO). La red no tiene una corporación dueña, por lo que no hay recolección masiva de datos personales centralizada.
- **Libertad de expresión y censura:** Permite libertad considerable, aunque cada pod puede tener reglas comunitarias. Algunos pods pueden moderar contenido no deseado (como material ilegal o discurso de odio), pero no hay un filtro global. Los usuarios pueden migrar entre pods si no les gusta la política de uno. En general, se considera resistente a la censura masiva que aplican empresas centralizadas (como Instagram).
- **Autenticación y anonimato:** El registro en un pod de Pixelfed es similar a cualquier servicio web: por email/usuario. Los usuarios pueden elegir seudónimo; no es obligatorio que vinculen su identidad real. Para la cuenta, normalmente se valida el correo con un enlace. Dado que cada servidor es independiente, la única “verificación” es la que el propietario del servidor decida (algunos exigen usuarios confirmados, otros no).
- **Fortalezas:** Ideal para compartir fotografías sin publicidad invasiva. Federado, por lo que no estamos atados a una sola compañía. Alto nivel de personalización de la experiencia según el servidor.
- **Limitaciones:** La creación de cuenta puede ser un obstáculo (hay que elegir un pod y confiar en sus admins). Para usuarios acostumbrados a algoritmos centralizados, puede parecer menos dinámico: el feed es cronológico (no hay algoritmo de recomendación en el protocolo). La calidad de la disponibilidad del servicio depende del pod elegido (los pods

pequeños pueden cerrar). Al no tener cifrado E2E, cualquier servidor federado puede mostrar o replicar el contenido público como quiera, aunque siempre cifrando el tráfico web.

2.3 Diaspora

Diaspora es una de las primeras redes sociales federadas (lanzada en 2010). Está diseñada con nodos llamados pods, que cualquier persona puede instalar. Cada pod mantiene la información de sus usuarios (perfiles, contactos, publicaciones). A nivel de protocolo, Diaspora utiliza un esquema de federación peer-to-peer propio (basado en diálogos JSON push/pull entre pods). Cuando un usuario publica “pistas” (posts), su pod replica ese contenido hacia los pods donde están sus seguidores.

La privacidad se controla mediante aspectos (grupos de contactos): un post puede etiquetarse para que solo ciertos aspectos lo vean. Técnicamente, los posts y comentarios se envían con firma digital (generalmente usando Ed25519) y cifrado en tránsito (TLS). Los pods sólo reciben datos según se haya compartido en común. Diaspora no cifra los posts de extremo a extremo (es más parecido a un Facebook descentralizado), pero cada pod es dueño de sus datos.

Para soberanía, esto significa que los usuarios eligen su instancia (pod) de confianza; pueden incluso crear una propia sin usar servidores externos. La red tolera la censura en tanto un pod puede decidir desconectarse de pods censores o públicos. Sin embargo, Diaspora depende del propietario del pod: un administrador puede leer los posts no compartidos públicamente. Aun así, su modelo de federación evita un punto único de falla y las políticas centralizadas de censura de una compañía grande.

Además, Diaspora no captura permanentemente metadatos: el contenido que no está compartido permanece solo en el pod local. Su política de privacidad es sencilla: no vende datos y al ser código abierto, cualquiera puede auditarlo. De esta forma, podemos considerar que Diaspora ejemplifica cómo una plataforma de red social puede repartir la autoridad entre sus miembros, reduciendo la vigilancia central, siempre que los administradores de pods actúen de buena fe.

- **Arquitectura y soberanía de datos:** Cada usuario elige o crea un *pod* (servidor). Los datos del usuario (posts, fotos, relaciones de seguimiento) se almacenan en su pod. El usuario es dueño de ese pod y puede decidir cuándo borrarlo o migrarlo. Esto brinda soberanía digital porque no hay un dueño único de la plataforma global; la comunidad autoorganiza los pods.

- **Federación:** Los pods de Diaspora se comunican entre sí (federan) usando un protocolo propio similar a ActivityPub (algunos info señalan que Diaspora pre-existe a ActivityPub pero usa un concepto de *"aspects"* y handshake para compartir posts). De esta manera, un usuario en Diaspora puede seguir amigos en otros pods de Diaspora, así como interoperar con otras redes (por ejemplo, permite seguir cuentas en Twitter desde Diaspora con plugins).
- **Privacidad:** Diaspora enfatiza la granularidad de la privacidad mediante las llamadas *aspects*: el usuario agrupa sus contactos (família, colegas, etc.) en aspectos, y decide qué publicaciones ve cada grupo. Esto permite compartir fotos privadas solo con aspectos definidos. Cada publicación puede tener settings especiales (pública, privada, en aspectos específicos). Los pods no leen el contenido privado de otras instancias (aunque técnicamente podrían interceptar datos al transmitirlos, la idea es confiar en la descentralización).
- **Datos y políticas:** Al no existir empresa central, las políticas dependen de cada administrador de pod. No hay acuerdos globales de monetización de datos. Cada pod se rige por su propio reglamento (por ejemplo, uno podría prohibir ciertas temáticas).
- **Libertad de expresión y censura:** Diaspora promueve altos niveles de libertad de expresión, con moderación distribuida. Si un podado impone censura, el usuario puede salir a otro pod. Esto hace muy difícil la censura generalizada. No obstante, al igual que Primal, hay riesgo de fragmentación en "burbujas" autocontenidas si los pods definen sus nichos.
- **Autenticación e identidad:** Cada usuario se identifica mediante un handle `@usuario@pod`. El pod puede requerir verificación de email, pero la identidad real no es verificada por la red; son seudónimos básicos. Los usuarios pueden tener varios perfiles en diferentes pods. No hay cifrado E2E de serie (Diaspora se enfoca en el control de quién ve el contenido, más que en ocultarlo, aunque se puede usar PGP en mensajes privados con mod).
- **Fortalezas:** Red de social abierta y descentralizada, pionera en el concepto de "tu(s) pod(s), tus reglas". Permite compartir texto, fotos, enlaces. La naturaleza federada evita monocultivos de contenido y monopolio de datos.
- **Limitaciones:** Los pods pueden variar en calidad y fiabilidad (algunos quedan abandonados). La interfaz y experiencia están más orientadas a nerds-techies que a público general; es menos plug-and-play que

plataformas centralizadas. No hay cifrado de extremo a extremo integrado, por lo que no es adecuada para discusiones sensibles (solo privacidad por grupos). Menor userbase que Mastodon o Facebook, lo que reduce el atractivo social para nuevos usuarios.

2.4 Mastodon

Mastodon es la red social federada más popular, inspirada en Twitter pero sin dueño. Usa el protocolo ActivityPub (de W3C). Cada usuario se registra en un "instance" (servidor Mastodon) que corre software libre en Ruby on Rails. Los posts ("toots") pueden ser públicos, solo seguidores o unicast a personas específicas. Cuando se crea un post, el servidor lo envía a otros servidores que tienen seguidores del autor, replicando así la red federada (similar a Diaspora, pero con estándar ActivityPub que facilita interoperabilidad con multitud de plataformas).

Desde el punto de vista de privacidad y vigilancia, Mastodon ofrece ventajas: los datos no están en manos de una corporación única; cada instancia define sus reglas (tiene poder local de moderación). Los usuarios pueden escoger instancias que no guarden registros profundos y que no entreguen datos a terceros. Tecnicamente, Mastodon elige TLS para cifrar conexiones bajo el modelo cliente-servidor. Sin embargo, la mayoría del contenido público no está cifrado (igual que Twitter). Por tanto, la comunicación pública es visible en texto claro, aunque propagada solo entre servidores participantes.

Mastodon favorece la soberanía digital al permitir la migración de usuario entre instancias (exportar y transferir su historial si decide mudarse). También posibilita bloquear y mutear a nivel de servidor o instancia entera, enfrentando la censura mediante el llamado "falseo de seguimiento" y el federado (un usuario censurado en una instancia puede unirse a otra).

Su protocolo ActivityPub promueve descentralización real: Mastodon puede integrarse con plataformas como Pleroma, PeerTube, PixelFed, etc., conformando un ecosistema interoperable. En síntesis, Mastodon ofrece comunicación en red sin autoridad central y la habilita técnicamente con especificaciones abiertas, minimizando la dependencia de entes corporativos, aunque deja en manos de cada servidor cómo se protege la privacidad de los datos no públicos.

- **Arquitectura y soberanía de datos:** Los usuarios se registran en *instancias* (servidores) Mastodon de su elección. Cada instancia almacena las publicaciones (toots) y datos de sus usuarios, siguiendo políticas propias. El usuario es dueño de su cuenta en ese servidor y controla sus

datos (puede exportar sus publicaciones, borrar cuenta, etc.). Dado que Mastodon es de código abierto, cualquiera puede lanzar su propio servidor y así tomar control total de la información de sus miembros.

- **Federación:** Usa el protocolo ActivityPub, permitiendo que usuarios de distintas instancias Mastodon e incluso otras redes ActivityPub (como PeerTube, Friendica, Pleroma) interactúen. Por ejemplo, un usuario en mastodon.social puede seguir a alguien en diabox.io (otro servidor Mastodon), y viceversa, como si fuera Twitter pero a nivel de servidores independientes. Esto amplía la red sin centralización.
- **Privacidad:** Por defecto, las publicaciones en Mastodon son públicas a menos que se marquen como privadas o para seguidores. Los usuarios pueden restringir sus publicaciones a solo seguidores aprobados. Dado el modelo descentralizado, cada instancia administra su propia base de datos: no existe recopilación de datos central. El protocolo ActivityPub cifra el transporte (HTTPS), pero las publicaciones llegan a los servidores de destino en claro. Para mensajería privada entre usuarios, Mastodon incluye mensajes directos (subsistema ActivityPub de mensajes privados), que no son cifrados E2E de base, aunque algunos usuarios han integrando soluciones manuales (PGP) fuera de la plataforma.
- **Moderación y censura:** La moderación en Mastodon es distribuida: cada instancia decide qué contenido es aceptable en su propio servidor (y qué usuarios bloquear o silenciar). Esto permite que comunidades autónomas creen reglas propias. La censura masiva es difícil porque no hay centralizador. Sin embargo, instancias muy estrictas pueden filtrar (por ejemplo, un servidor en un país opresivo podría bloquear contenido localmente). Mastodon incluye herramientas de publicación moderada (ocultar contenido sensible, bloquear hashtags) para mejorar seguridad de los usuarios.
- **Autenticación e identidad:** El registro típicamente requiere email y nombre de usuario (ejemplo: @usuario@instancia). Los usuarios pueden usar pseudónimos. Cada usuario tiene un “profile” básico con avatar, biografía, etc. La identidad está ligada a la instancia; si el servidor se cierra, el usuario puede migrar su cuenta o suscritsions a otro servidor.
- **Fortalezas:** Gran adopción, comunidad activa, interfaz atractiva. Su naturaleza abierta fomenta innovación (plugins, extensiones, apps móviles varias). Alta libertad: sin algoritmos, el timeline es cronológico. No hay que dudar de censura corporativa: la regulación es social/comunitaria.

- **Limitaciones:** La experiencia puede ser fragmentada: hay demasiados servidores para elegir, lo cual puede abrumar al usuario promedio. La variedad de políticas de moderación entre instancias puede crear confusión. No hay cifrado end-to-end para DMs por defecto, y la privacidad es frágil en DMs (depende de que ambos confíen en la instancia del otro).

2.5 Friendica

Friendica es una plataforma federada de microblogging y red social diseñada para interoperar con múltiples protocolos. Un servidor Friendica puede conectarse a redes ActivityPub (Mastodon, Pleroma, etc.), Diaspora, GNU-Social y hasta a Facebook/Twitter de forma unilateral. Su arquitectura es distribuida: cada usuario elige un servidor (o "nodo") donde abre su perfil. En cuanto a privacidad, Friendica pone énfasis en dar control al usuario sobre qué datos compartir.

Técnicamente, las conexiones servidor-servidor usan protocolos de fediverse estándar, y el cifrado se limita a TLS entre servidores. Friendica permite filtrar contenido y definir criterios complejos de visibilidad. Lo crucial es su compatibilidad "multiverso": alguien en Friendica puede seguir a un contacto en Mastodon o Pleroma sin salir de la misma interfaz, gracias a bridges.

Esto refuerza la independencia tecnológica, pues no obliga a todos a migrar a una sola plataforma; se diluye el riesgo de vigilancia al fragmentar comunidades entre distintos ecosistemas. Respecto a la resistencia a la vigilancia, Friendica no hace cifrado E2E propiamente (como el caso de DMs), pero como todos los mensajes pasan por los nodos federados, la confidencialidad depende de la confianza en cada servidor. No obstante, Friendica no requiere un real nombre ni email confirmado para registrarse, brindando cierta capa de anonimato en la superficie.

Los usuarios pueden configurar sus derechos de visibilidad (público, solo seguidores, grupos específicos), reduciendo exposición. En conjunto, Friendica actúa como un "hub" social federado: su valor para la soberanía reside en su flexibilidad federada multiplataforma y en empoderar a los usuarios para ejecutar su instancia, con políticas de privacidad propias. No es impenetrable contra vigilancia, pero su naturaleza fragmentada y multicanal dificulta la compilación global de datos por un solo ente.

- **Arquitectura y soberanía de datos:** Se instala en servidores (nodos) al igual que Mastodon. Los datos del usuario residen en el servidor donde está registrado. Permite publicar contenido (entradas de blog, estados,

fotos) que los contactos pueden ver. Cada instalación define su propia política de administración de datos.

- **Federación y compatibilidad:** Su característica principal es que "habla" varios protocolos: ActivityPub, Diaspora, Pump.io, GNU Social, RSS, etc. Esto significa que un usuario Friendica puede seguir tanto a usuarios de Mastodon (ActivityPub) como de Diaspora (pod), o incluso de Twitter (limitado a leer tweets si se enlaza la cuenta). Integra conexiones multi-plataforma. Friendica federado con Friendica, Mastodon y otros. Muy útil para unificar múltiples comunidades.
- **Privacidad:** Permite privacidad granular: un post puede ser público o visible solo por grupos particulares o por contactos. Friendica también puede integrarse con servicios de correo para notificar mensajes. El protocolo HTTPS asegura el acceso web. Por defecto, las comunicaciones se cifran de cliente a servidor y entre servidores. Para mensajes privados internos, Friendica no implementa E2E automático (se podría usar PGP en conexiones de email integradas, pero no es estándar).
- **Datos y políticas:** Cada servidor Friendica define reglas de datos. La plataforma no tiene una entidad central de monitoreo. Dado que Friendica puede hablar con redes comerciales (por ejemplo, conectarse con Twitter), el usuario debe confiar qué datos comparte cuando se usa esa opción.
- **Libertad de expresión:** Al ser federado con la Fediverso, comparte la misma ventaja de tolerancia y resistencia a la censura que Mastodon. Adicionalmente, dado que integra más redes, los usuarios pueden publicar en Friendica y que esa publicación se republique (los plugins permiten cross-posting), ampliando la audiencia sin depender de una empresa.
- **Autenticación e identidad:** Registro con email. Identidad similar a Mastodon (`@usuario@instancia`). Permite cambio de usuario o migración a otro server. Brinda también la opción de tener múltiples perfiles "sub-identidades" (como alias) en la misma cuenta.
- **Fortalezas:** Extremadamente flexible e interoperable. Ideal para usuarios con muchos círculos en distintas redes. Al federar ampliamente, no excluye a casi nadie.
- **Limitaciones:** Gran complejidad técnica. A nivel usuario puede resultar abrumador. La interfaz no es tan pulida como la de Mastodon; da sensación de blog o portal más que un chat social constante. El hecho de poder integrarse con redes centralizadas (por ejemplo, replicar tuits) implica confiar en esas redes externas si se usa. Requiere servidor propio o

de terceros (los que instalados consumidores son pocos, comparado con Mastodon).

2.6 PeerTube

PeerTube es una red social descentralizada enfocada en vídeos (como YouTube libre). Emplea ActivityPub: cada instancia PeerTube hospeda vídeos que los usuarios suben. Cuando un vídeo se publica, la instancia notifica a seguidores en otras instancias. Para la reproducción, PeerTube usa streaming descentralizado y BitTorrent: el vídeo puede descargarse parte por parte desde múltiples pares (otros usuarios que ya lo tengan) mientras se ve, reduciendo la carga en un único servidor. La arquitectura federada significa que cada comunidad controla sus propios vídeos y moderación. La política de privacidad es manejada a nivel de instancia, pero típicamente los vídeos son públicos por diseño.

Al usar tecnología P2P para distribución, PeerTube evita la centralización de ancho de banda; un gobierno interesado en censurar contenido no puede detener una vez que varios peers distribuyen fragmentos, salvo bloqueando direcciones IP específicas (lo que suele ser más difícil y se puede evadir con Tor). La federación de instancias también permite a los usuarios reenviar o replicar vídeos entre comunidades, manteniendo varios mirrors públicos. Desde la soberanía digital, PeerTube libera al creador de depender de plataformas propietarias: uno puede auto-hospedar su instancia o unirse a comunidades que respeten la privacidad.

No obstante, los metadatos (como listas de reproducción o estadísticas) residen en cada servidor, visibles para el admin, dado que no existe cifrado E2E en este contexto (las tablas de enrutamiento de torrents no están cifradas). A nivel institucional, PeerTube ofrece una capa de resistencia: no puede haber un apagón global tan sencillo como desconectar unos pocos centros de datos. Todo ello fomenta una cultura de vídeo libre y soberanía tecnológica, aunque la experiencia de usuario y la protección de datos dependerán de cada instancia y de las políticas que adopten sus administradores.

- **Arquitectura y soberanía de datos:** Cada servidor PeerTube (instancia) aloja sus propios videos, que los usuarios suben. El archivo multimedia está almacenado en el servidor del usuario que lo publicó. Los servidores pueden sincronizarse entre sí (federarse) para compartir videos mediante enlaces ActivityPub o directamente a través de torrents (p2p) cuando un usuario ve un video. El autor original controla su contenido, y cada

instancia decide cuánto ancho de banda destina para compartir con otras instancias.

- **Federación:** PeerTube usa ActivityPub: los servidores federan sus canales de video. Así, un canal en un servidor puede ser seguido desde otro. Cuando un usuario en una instancia A publica un video, los usuarios de instancias amigas (que siguen el canal o la etiqueta) pueden ver miniaturas y reproducirlo. Además, al abrir un video, el servidor del espectador se conecta a la red P2P (WebTorrent) para descargar fragmentos del video desde múltiples fuentes (servidores y otros espectadores), reduciendo la carga en el servidor original.
- **Privacidad:** Dado que PeerTube se concentra en contenido público, no hay mensajes privados de por medio. Sin embargo, las vistas/análisis básicos podrían registrarse localmente en cada servidor. No tiene tracking corporativo ni publicidad intrusiva (aunque se pueden integrar ads de forma local si el administrador lo desea). El acceso por HTTPS protege el streaming.
- **Censura y políticas de contenido:** Cada instancia gestiona su propio contenido: puede decidir permitir o prohibir ciertos videos, moderar comentarios, etc. La censura generalizada es difícil porque no hay una entidad central. Alguien podría sincronizarse con un canal en una instancia que censura y viceversa. Sin embargo, las instancias pueden bloquear otras si no quieren federar con ellas (por ejemplo, por hosting de contenido problemático).
- **Identidad y autenticación:** Los usuarios pueden registrarse (o anonimizarse) según el administrador del servidor. Cada instancia maneja sus cuentas. Pueden ser públicas o privadas. No hay verificación de identidad obligatoria, solo correo electrónico.
- **Fortalezas:** Plataforma verdaderamente distribuida; ideal para videoblogs, tutoriales o cualquier creador de contenido que no quiera estar sujeto a los algoritmos de YouTube. Los videos siguen disponibles siempre que al menos un nodo los comparta. Incentiva la redistribución colaborativa de ancho de banda.
- **Limitaciones:** La experiencia puede ser menos fluida (pero mejora continuamente). Los videos HD pueden no cargarse inmediatamente si hay pocos peers. Al principio, es necesaria paciencia porque no hay red masiva como YouTube. Los pods pequeños podrían saturarse con video de alta demanda. La interoperabilidad con otras redes sociales es menos directa (aunque se puede compartir enlaces manualmente). No hay gran

monetización incorporada; la comunidad promueve donaciones para apoyar instancias.

2.7 Pleroma

Pleroma es otra plataforma de microblogging federada, muy ligera comparada con Mastodon, pero usando el mismo protocolo ActivityPub. Su arquitectura esencialmente replica la filosofía de Mastodon: cada usuario reside en un servidor Pleroma, publica entradas (llamadas “toots” o “notes”) que se federan a otros servidores.

Sin embargo, Pleroma fue diseñada con eficiencia en mente: consume menos recursos, y usa Redis o bases de datos simples para almacenar eventos. En cuanto a privacidad, Pleroma comparte características similares a Mastodon: no hay cifrado E2E para mensajes públicos, solamente TLS en conexiones de cliente-servidor. La diferencia es su comunidad activa en instancias pequeñas y a menudo auto-hospedadas, lo que dispersa aún más la red social, dificultando la vigilancia macroscópica. Cada instancia puede elegir sus políticas (algunas no requieren registro formal o prometen no almacenar logs). Pleroma también incluye ajustes robustos de filtrado de contenido y bloqueo distribuido de usuarios, permitiendo a las comunidades autocontrolar contenido.

Desde la soberanía tecnológica, Pleroma ofrece una entrada más ligera al fediverso: en hardware modesto o incluso en la nube personal, cualquier persona puede montar su servidor. Esto amplía la capacidad de autodefensa: una comunidad perseguidos digitalmente puede rápidamente crear un nodo Pleroma y refugiarse allí. En resumen, Pleroma facilita el poder de estar “sobre tu propio servidor” con un costo mínimo, reforzando la resistencia a la censura y la autonomía en redes sociales.

- **Arquitectura y soberanía de datos:** Similar a Mastodon en términos de diseño federado. Los usuarios crean cuentas en servidores Pleroma, que almacenan sus publicaciones y relaciones. El software es de código abierto, libre, y bastante eficiente en recursos, lo cual lo hace apto para instancias pequeñas o de hardware limitado.
- **Federación:** Usa **ActivityPub** y es totalmente compatible con otras plataformas basadas en este protocolo (Mastodon, PeerTube, Friendica, etc.). Significa que un usuario Pleroma puede seguir y ser seguido por alguien en Mastodon o viceversa. Esto lo integra en el fediverso.
- **Privacidad:** Las publicaciones funcionan como en Mastodon: públicas por defecto o restringidas a seguidores. Los servidores cifran las conexiones

(HTTPS) y respetan la visibilidad marcada por el usuario. No hay cifrado de contenido (de nuevo, no es mensajería privada), pero los datos personales de usuarios no son explotados comercialmente, pues no hay entidad central con tal modelo. Cada instancia puede tener sus propias políticas de retención de datos.

- **Políticas y censura:** Puede adoptar cualquier política de contenido que decida el administrador. La capacidad de bloquear instancias facilita evitar contenidos no deseados (por ejemplo, un grupo de instancias librepensadoras puede bloquear las instancias homófobas, y viceversa). Esto crea redes virtuales de confianza.
- **Autenticación e identidad:** Registro con email/usuario. Identidad se maneja similar a Mastodon. Soporta alias de usuario (para diferenciar contextos) y la idea de "múltiples perfiles" bajo una cuenta (como modo privado/público).
- **Fortalezas:** Muy eficiente, requiere menos recursos de hardware (menos RAM/CPU) que Mastodon. Su dinamismo lo hace popular en instancias pequeñas. Interfaz moderna (hay apps móviles). La comunidad se enfoca en brindarle estabilidad y rapidez.
- **Limitaciones:** Al ser relativamente nuevo, tiene menos usuarios que Mastodon masivo. Algunas características faltantes en Mastodon sólo tardan en implementarse (por ejemplo, Pleroma tardó en tener búsqueda global, o algunas funciones de administración). La experiencia de usuario puede variar según la instancia (por ejemplo, qué temas bloquean). No ofrece cifrado E2E de DMs por defecto (igual que Mastodon).

2.8 Misskey

Misskey es una red social federada construida principalmente en Japón pero de alcance global, usando ActivityPub. Combina microblogging con funcionalidades avanzadas (reacciones gráficas, edición en tiempo real). Cada instancia Misskey opera de manera independiente; los usuarios pueden seguirse entre instancias y con otras compatibles (Mastodon, etc.).

En cuanto a modelo federado, Misskey implementa técnicas como la firma en los ActivityPub para autenticidad y mezcla de datos con otros servidores. Al ser software libre, permite a cualquiera lanzar su propia instancia. Respecto a privacidad, Misskey permite controles sobre la visibilidad de cada publicación y admite cifrado opcional para mensajes privados.

No obstante, su énfasis está más en características ricas (stickers, traducción automática, colecciones de contenidos) que en privacidad hardcore. En cuanto a resistencia a la vigilancia, Misskey posee varias opciones de UI para “silenciar” usuarios globalmente, incluso instancias enteras, ayudando a comunidades a protegerse del spam o de nodos maliciosos en la federación. Además, al ser menos conocido fuera de Japón, a veces pasa desapercibido para censores genéricos.

Desde la perspectiva de soberanía digital, Misskey es valioso porque promueve la diversidad de instancias y porque su código abierto permite auditoría independiente. En esencia, extiende la federación social con funcionalidades sociales modernas, conservando la filosofía de que el usuario decide dónde viven sus datos, lo que dificulta el seguimiento centralizado de personas en la red.

- **Arquitectura y soberanía de datos:** Al igual que Mastodon y Pleroma, los usuarios están en instancias independientes de Misskey. Cada servidor guarda las publicaciones (“notas”), “réactions” (emoji reaccion que a veces muestran quién reaccionó), imágenes, etc. Es de código abierto y permite a cualquiera ejecutar su propio servidor.
- **Federación:** También es compatible con ActivityPub, lo que lo integra en el fediverso. Un usuario de Misskey puede comunicarse con usuarios de Mastodon, PeerTube, etc. (aunque hay algunas peculiaridades en la integración debido a ciertos tipos de contenidos propios de Misskey, como las reacciones especiales).
- **Privacidad:** Ofrece configuraciones avanzadas por publicación: público, limitado a seguidores, etc. Además, incluye la función “anónimo en grupos” donde el autor puede publicar en un grupo de la instancia de forma anónima. Cada servidor escoge sus reglas de moderación. La naturaleza federada impide tracking masivo; el servidor sólo tiene datos de sus miembros.
- **Seguridad:** El tráfico web se protege con HTTPS. Misskey permite usar 2FA (authenticación de dos factores) para proteger cuentas. Para datos, almacena posts en texto plano (no cifrado).
- **Características únicas:** Misskey introduce reacciones con avisos en tiempo real (puede mostrar quién dio “like”). También incluye un sistema de *“drive”* personal, donde los usuarios pueden almacenar archivos privados. Tiene un calendario, mensajería interna, notas en trending y un diseño muy modular de perfil (widgets que el usuario puede editar).

- **Libertad de expresión:** Muy flexible; las instancias suelen ser autónomas en moderación. Existen instancias con diferentes políticas: algunas muy liberales, otras más censuradas (depende del país o comunidad). La federación asegura que no se puede censurar toda la red desde un solo lugar.
- **Autenticación e identidad:** Registro tradicional por email. Los usuarios tienen una URL como `usuario@instancia`. Pueden añadir perfiles extra ("sub-accounts") para actividades específicas.
- **Fortalezas:** Interfaz moderna, colorida, con énfasis social. Fuerte comunidad en Japón y creciente globalmente. Experiencia interactiva con "likes" y "reacciones" avanzadas. Versatilidad en uso (podría servir de blog, red social y gestor de archivos personales).
- **Limitaciones:** Curva de aprendizaje: tantas funcionalidades pueden confundir al usuario nuevo. Al igual que otras redes, si la instancia que elige cierra, debe migrar. El alto nivel de dinamismo (actualizaciones continuas) a veces puede introducir inestabilidades. Como su ecosistema es más pequeño, no tiene la masa de usuarios de Mastodon, pero se está expandiendo lentamente fuera de Japón.

3. Políticas y aparatos de vigilancia gubernamentales – Análisis crítico

Las tecnologías enumeradas en las secciones anteriores potencian la privacidad individual. Sin embargo, los gobiernos y agencias de inteligencia también despliegan herramientas y legislaciones para la vigilancia masiva. Esta sección examina críticamente algunas prácticas y leyes relevantes.

3.1 Ley 21.821 sobre el Sistema de Inteligencia del Estado (Chile)

La Ley N°21.821 (Chile, 2026) reforma la legislación de inteligencia nacional para "fortalecer y modernizar" las capacidades estatales. En la práctica, amplía drásticamente las facultades de agencias de inteligencia y defensa, afectando severamente la privacidad de las comunicaciones y datos personales. Entre sus cambios clave está la definición de inteligencia como "obtención y análisis sistemático de datos e información para alertas", lo que abarca implícitamente cualquier medio de recolección masiva (sin requerir sospecha previa). La ley crea también la Agencia Nacional de Inteligencia y refuerza la coordinación de los cuerpos militares y policiales en actividades de vigilancia.

Su cláusula más alarmante es el Artículo 5° bis (nuevo): obliga a todos los órganos públicos, estatales y entidades con participación estatal a **entregar inmediatamente cualquier antecedente, dato personal e información solicitada** por los jefes de inteligencia. Esto incluye ministerios, municipalidades, empresas públicas, e incluso privados consultados por estos entes. La única excusa para negar información sería el secreto absoluto ("seguridad nacional"); en la práctica abre una puerta a compartir bases de datos estatales (de salud, policía, tributos, etc.) a discreción de las autoridades de inteligencia. Además, la ley autoriza el intercambio de datos personales entre organismos de inteligencia previo a autorización interna, lo que implica que comunicaciones interceptadas o registros de actividad pueden circular libremente dentro del sistema, reduciendo la confidencialidad al mínimo.

En conjunto, la Ley 21.821 crea un entorno legal donde las instituciones de inteligencia pueden realizar vigilancia amplia sin contrapesos externos reales. Esto choca con artículos constitucionales de inviolabilidad de la comunicación y la protección de datos. En términos de seguridad electrónica, significa que un detective de inteligencia puede orden informáticamente la recolección de emails, registros telefónicos, datos de geolocalización, expedientes bancarios, etc., sin judicializarlo.

El derecho a la integridad y confidencialidad de las comunicaciones queda suprimido de facto: la ley delega la "autorización" de recolección a autoridades ejecutivas, no a jueces independientes. En la práctica, se temen interceptaciones indiscriminadas de comunicaciones y activación de DPI o hackeos legales ordenados bajo esta cobertura. Desde la perspectiva de autodefensa, esta reforma obliga a los ciudadanos conscientes a extremar medidas criptográficas y anónimas: ahora la ley avala cualquier medio de vigilancia para fines de "seguridad", vulnerando la autenticidad (firmas digitales, cifrados privados) al no ofrecer garantías externas ni transparencia real.

- **Objeto y alcances:** La ley define las atribuciones de la Agencia Nacional de Inteligencia (ANI) y otros organismos, estableciendo que pueden recolectar y analizar información estratégica. Sin embargo, la redacción es amplia y no detalla estrictamente límites claros, lo que abre la puerta a interpretaciones extensivas.
- **Vulneración de la privacidad:** Se señala que la ley faculta a autoridades a interceptar comunicaciones y acceder a datos sin orden judicial previa clara en todos los casos. Esto contraviene principios de un debido proceso. Al permitir intervenciones "indiscriminadas y sistemáticas", viola derechos

a la **integridad, confidencialidad y autenticidad** de las comunicaciones personales. Por ejemplo, sería posible espiar llamadas, correos electrónicos o mensajes sin consentimiento, al amparo de la ley.

- **Falta de supervisión y garantías:** Un problema destacado es la insuficiente fiscalización externa. Los organismos de inteligencia podrían operar sin rendir cuentas ante entidades independientes. La falta de mecanismos de control ciudadano (como supervisión parlamentaria efectiva) implica un riesgo de abuso de poder. Históricamente, sistemas similares permitieron espionaje político o social sin consecuencias legales.
- **Derechos fundamentales en riesgo:** La ley pone en riesgo la libertad de expresión y el derecho a la privacidad. Al saber o temer que sus comunicaciones pueden ser intervenidas, periodistas, activistas y ciudadanos podrían autocensurarse. El principio de confidencialidad de la comunicación, protegido por la constitución, se debilita frente a una norma tan amplia. La opacidad en las operaciones de inteligencia contrasta con la transparencia necesaria en un Estado de derecho.
- **Implicaciones prácticas:** En la práctica, la ANI y otras entidades podrían utilizar esta ley para implementar tecnologías de vigilancia masiva (como intercepción de metadatos telefónicos, análisis de redes sociales, etc.) con muy poca restricción legal. Por ejemplo, podrían acceder a bases de datos de información personal (historial bancario, ubicación geográfica, tráfico en redes sociales) supuestamente para “seguridad”, pero con riesgo de persecución de opositores o control social indiscriminado.
- **Críticas desde la sociedad civil:** Organismos de derechos humanos han criticado la ley por desalentar el pluralismo y la privacidad ciudadana. En contraste, los defensores argumentan que la ley solo formaliza y regula técnicas de espionaje ya existentes (como interceptación de comunicaciones). La clave crítica es que esa regulación debería equilibrarse con fuertes garantías, las cuales se perciben débiles o nulas en la ley actual.
- **Conclusión crítica:** En resumen, la Ley 21.821, aunque buscaba actualizar el marco legal de inteligencia, termina otorgando poderes amplios a las autoridades en detrimento de libertades fundamentales. El análisis concluye que requiere revisión urgente: se necesitan salvaguardas claras (órganos independientes de control, transparencia mínima, límites estrictos) para evitar abusos y proteger la integridad de las comunicaciones de los ciudadanos.

3.2 Inspección profunda de paquetes (DPI)

La **inspección profunda de paquetes (DPI)** es una tecnología de monitoreo de redes que examina el contenido completo de cada paquete IP, no solo sus encabezados. Técnicamente opera en los niveles superiores del modelo OSI (hasta capa de aplicación), leyendo los datos transportados. Con DPI, una entidad puede identificar protocolos, extraer contenidos (como encabezados HTTP, cuerpos de email, fragmentos de video o chats) y aplicar reglas de filtrado muy precisas. Desde el punto de vista de la privacidad, DPI es intrusivo porque puede violar la confidencialidad: por ejemplo, un Gobierno puede capturar paquetes de un túnel TLS (si posee la llave privada) o, más común, reconocer patrones de tráfico incluso cuando viajan cifrados.

Los dispositivos DPI pueden detectar cosas como qué páginas web visitamos, qué servicios usamos (Telegram, Signal, WhatsApp), e incluso contenidos en mensajes no cifrados o metadatos detallados (p.ej., duración de una llamada VoIP). En manos estatales, DPI permite vigilancia masiva a gran escala: los gobiernos despliegan DPI en peering points de internet, ISP o fronteras nacionales para analizar todo tráfico. Esto se usa para censurar (bloquear sitios web, VoIP, redes sociales) o espiar (seguir patrones de comunicación, localizar opositores). Por ejemplo, China utiliza DPI para su Gran Cortafuegos, identificando y bloqueando VPN o servicios disidentes; otros países usan DPI para filtrar contenido ilegal o interceptar mensajería de determinados protocolos. El efecto es que la **intimidad electrónica queda comprometida**: incluso si el contenido del paquete no es legible (debido a cifrado), DPI puede revelar con quién hablas, con qué frecuencia y a qué hora, socavando la integridad y confidencialidad de los intercambios.

Además, DPI puede insertar respuestas (ataques Man-in-the-Middle) si opera con la autoridad suficiente: por ejemplo, redirigir tráfico a servidores de vigilancia, introducir certificados falsos, o degradar la encriptación permitiendo capturar sesiones HTTPS. En resumen, la implementación de DPI por parte del Estado mina el secreto de comunicaciones al magnificar la capacidad de análisis de datos: un simple router con DPI activo puede mantener archivos de registro exhaustivos de lo que hacen los usuarios en la red. Como contramedida, el uso de cifrados fuertes e incluso técnicas de evasión (obfuscación de tráfico, Tor, VPN seguras con cifrado resistente) es imprescindible para preservar la privacidad ante este tipo de inspección gubernamental.

- **Cómo funciona:** DPI opera al analizar los *paquetes* de datos que se envían por la red. A diferencia de la inspección tradicional que sólo verifica la cabecera (direcciones IP, puertos, etc.), DPI inspecciona la carga útil

(payload). De este modo, puede identificar protocolos específicos (por ej. HTTP, P2P, VoIP) y extraer información contextual (palabras clave, URLs, tipos de archivos). Esto lo hace potencialmente tan invasivo como "descifrar" el contenido, incluso cuando la conexión está cifrada parcialmente, ya que busca patrones o comportamiento para su clasificación.

- **Usos gubernamentales:** Muchos gobiernos emplean DPI para **vigilancia y censura**. Ejemplos: filtrar sitios web (mediante bloqueo de paquetes que contengan ciertas direcciones), monitorear descargas de torrent, detectar uso de VPN o Tor (bloqueando los protocolos conocidos), recopilar metadatos (frecuencia de comunicación entre IPs). También se usa en la lucha contra amenazas (terrorismo, crimen organizado), pero el riesgo es su uso generalizado sin orden judicial.
- **Impacto en privacidad:** DPI puede revelar información sensible: si un usuario envía un correo no cifrado, DPI puede leer su contenido; incluso con HTTPS, puede identificar los servidores consultados (Hostnames) y en algunos casos inferir patrones de uso. Se convierte en una forma de **vigilancia digital de masas**: todos los paquetes son analizados. Esto atenta contra la confidencialidad de las comunicaciones. Además, DPI puede ser utilizado para crear perfiles de navegación y comportamiento de cada usuario (por ejemplo, interés en ciertos temas, frecuencias de conexión).
- **Vigilancia y censura:** Gobiernos usan DPI para **bloquear servicios** (p.ej. impedir el acceso a redes sociales, sitios de noticias, plataformas de mensajería). También pueden introducir desinformación (por ejemplo, inyectando contenido falso en páginas visitadas). Por su capacidad de discriminar tráfico, DPI puede ralentizar o degradar servicios considerados indeseables. En regímenes autoritarios, es común para silenciar medios disidentes o restringir protestas online.
- **Legalidad y ética:** En muchos lugares, el uso de DPI plantea dilemas legales: según legislaciones de telecomunicaciones, DPI puede incumplir normas de neutralidad de red (trato no equivalente). Además, sin salvaguardas, viola derechos humanos internacionales sobre privacidad. Aunque en algunos países se justifica en la ley como herramienta anti-crimen, generalmente la ejecución no está transparentada, siendo una 'caja negra' para la sociedad.
- **Ejemplos de implementación:** DPI no es exclusivo de gobiernos: puede usarse en empresas (para gestión de red), o incluso proveedores de contenido (e.g., Netflix midiendo calidad). Sin embargo, el análisis crítico se centra en **gobiernos** y grandes ISP que cooperan con ellos. Estudios han

documentado la instalación de equipos de DPI en puntos de intercambio de internet en China, Irán, Rusia y otros, con alcance nacional.

- **Respuesta ciudadana:** Frente a DPI, surgen herramientas de resistencia: uso de VPNs, Tor, cifrado extremo-a-extremo en aplicaciones (que hace difícil el análisis de contenido), protocolos ocultos (mejoras para saltarse las detecciones de DPI). No obstante, gobiernos reaccionan bloqueando o forzando el uso de DPI para interceptar incluso protocolos cifrados mediante esquemas como *SSL stripping* (dejando caer conexiones a HTTP) o introduciendo certificados falsos (mitm).

En suma, la inspección profunda de paquetes es una tecnología poderosa que puede fácilmente *violentar la privacidad del usuario*. Su uso indiscriminado por gobiernos constituye una amenaza grave a la integridad de las comunicaciones y a la libertad digital. La sociedad y la legislación deben exigir transparencia y límites claros para cualquier aplicación de DPI, así como promover contramedidas (como el cifrado robusto) para salvaguardar los derechos fundamentales.

3.3 Vigilancia en infraestructuras de telecomunicaciones

Los gobiernos también emplean otras metodologías de vigilancia dentro de las infraestructuras de telecomunicaciones que van más allá de DPI, afectando la integridad, confidencialidad y autenticidad de la información:

- **Intercepción de comunicaciones:** Mediante mandatos legales o extrajudiciales, las autoridades pueden ordenar a proveedores de servicios (ISP, operadores móviles) que instalen equipos de escucha. Por ejemplo, monitorear llamadas telefónicas o chats en tiempo real. Esto compromete la confidencialidad: los usuarios creen estar en una conversación privada, pero pueden ser grabados sin su conocimiento.
- **Metadatos de comunicaciones:** Aun cuando no se escuchan contenidos, recolectar metadatos (quién se comunicó con quién, cuándo, cuánto duró la llamada o tamaño del mensaje) permite perfilar usuarios. Sistemas de inteligencia masiva como los desarrollados por la empresa Palantir, que los estudiaremos y analizaremos en detalle más adelante, se alimentan de esos metadatos para inferir relaciones y comportamientos.
- **Redes sociales y mensajería:** Se pueden obligar a empresas a facilitar (backdoors) o registrar chats de apps populares. Por ejemplo, presionar a empresas a entregar llaves de cifrado, o a transmitir a las fuerzas de

seguridad comunicaciones cifradas (un debate constante sobre cifrado end-to-end).

- **Hackeo directo y malwares estatales:** Aparte de interceptar a nivel de red, los estados emplean malware para espiar directamente en dispositivos de alto interés (ver sección de Pegasus abajo). Esto evita incluso la necesidad de interceptar la comunicación en tránsito: extrae la información en el origen (cuando está descifrada).
- **Integridad de las comunicaciones:** Cuando el gobierno puede inyectar contenido (por ejemplo, modificar paquetes en tránsito) cuestiona la **autenticidad**: el receptor no sabe si el mensaje que llega es genuino o fue alterado. DPI combinado con técnicas de inyección permite suplantar páginas o insertar malware (ataques man-in-the-middle). Esto afecta la confianza que un ciudadano puede tener en las redes de telecomunicación como fuente fiable.
- **Bloqueo y desconexión:** En casos extremos, se han documentado cierres de redes móviles o internet (para evitar la difusión de protestas). Esto implica una censura total, afectando la libertad de información y de reunión.
- **Contexto local:** En Chile y otros países de la región, hay debates sobre cómo se aplican estas técnicas. Se han denunciado prácticas de vigilancia de comunicaciones políticas. La posibilidad de usar DPI y otros mecanismos sin supervisión judicial aumenta el riesgo de espionaje interno de opositores, defensores de derechos humanos o periodistas.
- **Conclusión:** La vigilancia en infraestructuras de telecomunicaciones es muy invasiva y puede violar múltiples derechos básicos. La clave es la falta de transparencia y control en la forma en que se aplica. La ciudadanía debe estar alerta y exigir límites legales claros, así como promover la adopción de herramientas que garanticen cifrado verdadero (como los mensajeros seguros revisados en la sección 1). Solo así se preserva la integridad (que el mensaje enviado llegue intacto), la confidencialidad (que terceros no puedan leerlo) y la autenticidad (que sabemos quién envió qué) de nuestras comunicaciones.

3.4 Impacto sobre la privacidad y derechos fundamentales

Las políticas y tecnologías de vigilancia discutidas tienen impactos directos en los derechos humanos y la autodeterminación digital:

- **Integridad de las comunicaciones:** Cuando las autoridades pueden intervenir o modificar nuestras comunicaciones sin consentimiento ni

conocimiento, la integridad (que el contenido no sea alterado maliciosamente) se resiente. No hay garantía de que la información recibida sea la que envió la otra parte originalmente. Esto rompe la confianza en el canal de comunicación.

- **Confidencialidad:** Como punto central, la vigilancia gubernamental ataca la confidencialidad de las comunicaciones privadas. La capacidad de cifrado fuerte de las aplicaciones libres (señalada en la sección 1) busca contrarrestar esto. Sin embargo, leyes permisivas y tecnologías invasoras minan este beneficio, porque pueden forzar la extracción de datos antes de cifrarlos o romper el cifrado en el backend.
- **Autenticidad:** La vigilancia masiva y manipulación de información amenaza con suplantación de identidad. Si un intermediario (por ej. un router estatal con DPI) puede inyectar o suplantar mensajes, el receptor no tiene forma fácil de verificar que el remitente es legítimo. Esto es grave en contextos políticos: por ejemplo, difundir noticias falsas que simulen provenir de fuentes oficiales.
- **Libertad de expresión:** El miedo a la vigilancia genera autocensura. Activistas o ciudadanos pueden abstenerse de expresarse libremente en redes sociales o chats por temor a represalias. La existencia de leyes de vigilancia sin controles intactos crea un efecto disuasorio.
- **Derecho a la privacidad:** Todo lo anterior se agrupa en la violación del derecho fundamental a la privacidad. En sociedades democráticas, esperaríamos que comunicaciones privadas se respeten a menos que exista causa judicial. Pero las dinámicas actuales muestran que la línea de intervención se difumina. Las comunicaciones en canales supuestamente privados (mensajerías cifradas, correos personales) pueden ser monitoreadas mediante orden judicial o incluso extrajudicial, consumando una vulneración de la privacidad personal y colectiva.
- **Soberanía digital:** En un plano más amplio, estas prácticas afectan la soberanía tecnológica. Si dependemos de infraestructuras telecom gobernadas o vigiladas, perdemos control sobre nuestra información. El empoderamiento mediante apps libres y protocolos descentralizados (visto en la sección 1 y 2) es una respuesta: al diversificar la red y difuminar el control, es más difícil para un gobierno central intervenir a gran escala. Sin embargo, sin un cambio legal o mayor conciencia pública, la lucha es cuesta arriba.
- **Necesidad de contrapesos:** El análisis crítico resalta que la sociedad debe exigir contrapesos legales (auditorías independientes, límites claros de

recolección, transparencia en la adquisición de herramientas de vigilancia). Además, es imperativo que las instituciones promuevan la educación digital: que la gente entienda qué derechos están en juego y cómo protegerse.

En suma, la íntima relación entre la tecnología de vigilancia y las libertades civiles exige vigilancia ciudadana. Toda política de seguridad que desatienda la protección de la integridad, confidencialidad y autenticidad de las comunicaciones, está erosionando pilares democráticos fundamentales. El conocimiento y difusión de las herramientas libres presentadas en este informe son parte de esa resistencia: empoderar a los usuarios con medios para resguardar sus derechos en el ámbito digital.

4. Malware y sistemas de espionaje avanzados

El panorama de amenazas digitales se complejiza con el uso de malware sofisticados por parte de gobiernos y actores estatales. En esta sección analizamos algunos de los más notorios, su tecnología subyacente y cómo defenderse.

4.1 Pegasus (NSO Group)

Pegasus es el nombre del sofisticado malware desarrollado por la empresa israelí NSO Group. Se trata de un **spyware militar de grado extremo** que infecta teléfonos móviles iOS y Android de forma furtiva. Técnicamente, Pegasus opera como una suite de exploits zero-day: utiliza vulnerabilidades inéditas en aplicaciones estándar (como iMessage de Apple, WhatsApp, o navegadores) para instalar código malicioso sin interacción del usuario (“zero-click”). Por ejemplo, se han identificado cadenas de exploits que aprovechan bugs en el motor de procesamiento de imágenes y llamadas de WhatsApp. En iPhones, Pegasus puede llevar a cabo un “jailbreak” parcial en memoria, quebrando las protecciones del sistema, y luego incorporar su payload. En Android, busca privilegios de root o solicita permisos de alta confianza si no logra escalarlos. Una vez dentro, Pegasus se mantiene oculto y puede **auto-destruirse** para eliminar rastros si sospecha que será descubierto.

En cuanto a capacidades, Pegasus convierte el móvil en un dispositivo completamente comprometido: lee SMS y correos, intercepta llamadas, registra ubicación GPS, activa micrófono y cámara, extrae credenciales y archivos (fotos, mensajes, contactos). Incluso puede monitorear aplicaciones de mensajería cifrada; por ejemplo, graba el texto de mensajes en pantalla antes de cifrarlos o captura contraseñas almacenadas. Se conecta periódicamente a servidores de

comando y control (C&C) manejados por el cliente gubernamental de NSO para descargar instrucciones y exfiltrar datos. Pegasus implementa técnicas avanzadas de sigilo: patchworks de cifrado interno, comunicaciones enviadas en túneles cifrados para evitar detección, timestamps y logs alterados para frustrar análisis forense. Además usa un “Triple Ratchet” propio para las claves internas, asegurando forward secrecy de la propia sesión de espionaje. Los métodos de infección incluyen tanto ingeniería social (SMS con enlaces maliciosos) como ataques sin interacción (cadenas de exploits en apps).

Los gobiernos adquieren licencias de Pegasus supuestamente para terrorismo, pero se han documentado usos para espiar periodistas, disidentes y opositores políticos en decenas de países. Por ejemplo, Citizen Lab y otros han encontrado instalaciones en smartphones de activistas en México, Arabia Saudita, India, entre otros. El software permite a los operadores extraer grandes volúmenes de datos personales e inteligencia de forma encubierta. Para contrarrestarlo, es imprescindible mantener el sistema operativo siempre actualizado (cerrando vulnerabilidades conocidas), desactivar vistas previas en apps de mensajería, y vigilar indicadores de compromisos (por ejemplo mensajes de error raros). Sin embargo, dado el nivel ‘militar’ de Pegasus, la defensa completa es muy difícil. Su existencia ejemplifica cómo los Estados pueden penetrar teléfonos móviles sin límites formales, violando la confidencialidad e integridad de las comunicaciones personales y los datos más íntimos.

A continuación se detalla su funcionamiento general:

- **Vector de ataque:** Pegasus ha evolucionado desde ataques de “clickbait” (requiriendo que la víctima abra un link malicioso) hasta **ataques de zero-click** (que no requieren acción del usuario). Los vectores más famosos incluyen exploits en apps de mensajería como WhatsApp (2019) o iMessage (2021), donde bastaba con recibir un mensaje especial para infectar el dispositivo sin necesidad de interacción. Aprovecha vulnerabilidades *día cero* (no conocidas públicamente) para instalar su carga.
- **Capacidades de vigilancia:** Una vez instalado en el dispositivo (Android o iOS), Pegasus obtiene **acceso root** o privilegios máximos. Esto le permite:
- Leer mensajes de texto, correos, chats cifrados (los extrae antes de ser cifrados).
- Grabar llamadas telefónicas y de apps (audio y video).
- Activar remotamente micrófono y cámara.
- Extraer credenciales, contraseñas y llaves criptográficas.
- Rastrear la ubicación GPS en tiempo real.

- Recopilar historiales de navegación y archivos almacenados.
- Persistir mediante técnicas anti-resete (sobrevivir a reinicios).
- **Técnicas de instalación y ocultamiento:** Pegasus ha incluido *"módulos"* según las necesidades del cliente (gobierno):
- Módulo de bypass para iPhones (aprovechando exploits de iOS).
- Módulo de Android usando vulnerabilidades de kernel.
- Uso de algoritmos de **steganografía** para ocultar la comunicación con sus servidores comando/control.
- Eliminación remota: el atacante puede borrar todos los rastros de la infección (auto-destruct mode).
- Auto actualizaciones encubiertas: para extender la operación sin que el objetivo se percate.
- **Impacto y ejemplos reales:** Casos célebres de Pegasus incluyen el espionaje a defensores de derechos humanos y periodistas del consorcio Forbidden Stories (informe Pegasus, 2021), así como espionaje a figuras políticas en múltiples países. Se estima que ha afectado a decenas de miles de víctimas globalmente.
- **Crítica técnica:** Pegasus es el ejemplo de cómo un solo malware puede desafiar toda la seguridad de un smartphone promedio. Su sofisticación técnica incluye ingeniería inversa y explotación profunda del sistema operativo.
- **Implicaciones para la privacidad:** Este tipo de spyware demuestra que, incluso usando apps seguras con cifrado fuerte, un objetivo clasificado puede ser interceptado **antes** de que la seguridad del contenido entre en acción (es decir, en el origen). Así, la integridad y confidencialidad de las comunicaciones queda comprometida vía dispositivo.
- **Medidas de detección y mitigación:** Dada la naturaleza clandestina de Pegasus, detectarlo es complejo. Herramientas especializadas (p.ej. Mobile Verification Toolkit de Amnistía) intentan analizar patrones en los logs de teléfono. Sin embargo, el promedio de usuario no cuenta con medios para comprobarlo. Por ende, se recomiendan:
- **Actualización constante:** Tener el sistema operativo siempre a la última versión (iOS/Android) es crítico. Muchas vulnerabilidades explotadas son parches pendientes. Mantener el SO y todas las apps actualizadas cierra muchas de las puertas de entrada.
- **Precaución con links/mensajes:** Aunque Pegasus puede ser 0-click, reducir las interacciones con contenido sospechoso (links, archivos de dudosa procedencia) disminuye riesgos de versiones menos avanzadas.

- **Entornos seguros:** En ambientes de alto riesgo, usar dispositivos “limpios” dedicados (teléfonos con firmware de seguridad, sin instalar apps extras) puede ayudar. Algunos recomiendan usar sistemas operativos móviles que dificulten la persistencia de malware (ej. sistemas especializados o incluso móviles con salvaguardas de hardware).
- **Restauración periódica:** En el caso de sospecha de infiltración, hacer restablecimiento completo de fábrica (aunque Pegasus muy avanzado podría persistir en ciertas capas, generalmente elimina su presencia).
- **Uso de apps seguras:** Aunque no protege contra Pegasus, usar mensajerías cifradas previene que, tras la infección, el malware robe contenidos de chats no seguros. No confiar en apps poco seguras.
- **Conclusión:** Pegasus representa la amenaza estatal máxima al dispositivo móvil personal. Su análisis técnico revela un salto tecnológico en espionaje. Para la resistencia electrónica, la clave es la prevención y la higiene digital: minimizar exposición de vulnerabilidades y estar consciente de los riesgos. Aún así, la única defensa infalible ante un adversario con recursos estatales puede ser la discreción tecnológica (no almacenar información extremadamente sensible en dispositivos vulnerables).

4.2 Palantir y la vigilancia masiva

Palantir es una empresa estadounidense que desarrolla tecnologías de análisis de macrodatos para inteligencia, vigilancia y seguridad. Con estas herramientas, es capaz de consolidar datos de múltiples fuentes en un único repositorio analítico. Arquitectónicamente, Palantir ofrece entornos centralizados (por ejemplo Palantir Gotham para agencias de inteligencia) que pueden ingerir registros de llamadas, actividad bancaria, imágenes de satélite, metadatos de redes sociales, DNS logs, datos biométricos, etc. Técnicamente, reúne estas fuentes heterogéneas usando APIs, conectores especiales y bases de datos híbridas (relacionales, grafos y NoSQL). Una vez importados, Palantir normaliza y “enriquece” los datos (cruce de tablas), permitiendo a analistas humanos crear gráficos de conexiones entre personas, lugares y eventos. Utiliza algoritmos de machine learning para detectar patrones o prever amenazas. Su interfaz es principalmente gráfica y basada en consultas interactivo, con capacidades de filtrado, zoom geoespacial y timeline.

En términos de vigilancia, Palantir permite vincular datos privados a identidades conocidas. Por ejemplo, puede fusionar la información telefónica de una persona con sus transacciones crediticias y sus publicaciones públicas en redes sociales, pudiendo construir un perfil exhaustivo y avanzado sobre toda su vida, sus

relaciones personales y actividades del día a día. Con estas tecnologías, Palantir extrae y procesa todo tipo de información, generalmente obtenida de la vigilancia masiva de telecomunicaciones, camaras de seguridad, escuchas telefónicas, redes sociales, imágenes satelitales y bases de datos empresariales y gubernamentales. Todas esas enormes cantidades de información se utilizan para realizar análisis de inteligencia de datos con los cuales desarrollar modelos de control social y manipulación de masas. Con este nivel de información es posible cambiar y redirigir el curso de un país al gusto y placer de algún sector económico o político. De esta forma, convierten las vidas íntimas y las relaciones sociales de millones de personas de continentes enteros en un producto a la venta para gobiernos y otras empresas.

Para conseguir este objetivo, se aprovecha de la centralización del Estado: datos que antes estaban aislados en «silos» (policía, migración, tributación) ahora se combinan fácilmente. Esto implica una erosión de la privacidad colectiva: un analista puede descubrir, mediante una sola plataforma, vínculos que antes no eran obvios. Por ejemplo, la policía podría identificar redes criminales rastreando comunicaciones, chats, viajes, pagos o movimientos entre sus miembros. Sin embargo, esto también se utiliza para rastrear y vigilar a grupos activistas, partidos políticos, movimientos sociales y organizaciones diversas con el único objetivo de clasificarlos, vigilarlos, controlarlos y anticiparse a cualquier desenlace de eventos o cadena de circunstancias diversas que puedan de alguna manera influir en el orden social y status quo de una nación. Asímismo, agencias de inmigración han usado Palantir para buscar personas indocumentadas cruzando fronteras mediante cruces de cámaras y bases de datos de viajeros.

Desde un punto de vista técnico, Palantir opera en infraestructuras robustas (nube privada o datacenters protegidos). Aunque la empresa afirma que los datos están bajo control del cliente, en la práctica la escala es tal que puede procesar big data en tiempo real. Como contramedida, la protección ante Palantir consiste en la fragmentación de la información (uso de cifrados de extremo a extremo en comunicaciones e incluso en bases de datos gubernamentales, siempre que sea posible) y en la supervisión pública del uso de estas herramientas. En síntesis, Palantir es el ejemplo de cómo la vigilancia moderna ya no ocurre solo interceptando datos en tránsito sino consolidándolos y analizándolos con IA: permite a los Estados “espiar” retrospectivamente a las personas utilizando cualquier dato almacenado digitalmente y conectándolo de forma inteligente con muchos otros registros, conexiones y relaciones vinculados con el objetivo, representando así un ataque y una violación a la privacidad así como una amenaza directa a la vida íntima de las personas.

A continuación se describen sus características más relevantes:

- **Funciones principales:** Palantir compila datos de múltiples fuentes (bases de datos gubernamentales, registros financieros, redes sociales, información de inteligencia, etc.) y usa algoritmos de inteligencia artificial/machine learning para detectar patrones, redes de relaciones o anomalías. Por ejemplo, puede crear perfiles de personas, visualizar conexiones de comunicaciones o transacciones, y predecir posibles amenazas.
- **Ejemplos de uso:** En el sector gubernamental, Palantir se ha empleado para:
- **Seguimiento de crimen organizado**: analizando vínculos financieros y de comunicaciones entre sospechosos.
- **Proyectos de antiterrorismo**: combinando información de vigilancia aérea, faciales en video y bases de datos migratorias.
- **Casos de inmigración**: podría filtrar migrantes en base a diversos indicadores.
- Durante la pandemia COVID-19 en algunos países, Palantir fue usado para rastrear contactos e información logística (controvertido por centralizar datos de salud).
- **Vigilancia de datos personales:** Al centralizar datos dispersos (teléfono, correo, redes sociales, ubicación GPS), Palantir crea un perfil extremadamente completo de individuos. Esto incluye historial de ubicaciones, relaciones sociales, creencias (si se analiza contenido en redes sociales), e incluso predecir conductas futuras. Tal agregación masiva de información plantea graves riesgos de privacidad: datos que por separado podrían ser inofensivos se vuelven muy sensibles al combinarse.
- **Capacidades de espionaje:** No es un spyware instalado en dispositivos; funciona a nivel institucional. Sin embargo, sus efectos son similares a ver *todo un Gran Hermano digital*. Por ejemplo, un gobierno con Palantir puede "escanear" comunicaciones públicas (los posts en redes), combinarlos con escuchas telefónicas y transacciones bancarias para reconstruir el perfil integral de un individuo.
- **Implicaciones sociales:** El uso de Palantir por fuerzas policiales ha recibido críticas: se cuestiona la falta de transparencia y supervisión. Al implementar vigilancia predictiva, existe el riesgo de detenciones basadas en análisis de datos más que en actos concretos, lo cual puede llevar a discriminación o persecución injusta. Además, dado que Palantir tiene contratos internacionales, la misma herramienta podría caer en manos de

regímenes autoritarios, exportando modelos de vigilancia de un país al otro.

- **Contra Poder y oposición:** Grupos defensores de la privacidad y empresas tecnológicas alternativas claman por contramodelos: por ejemplo, aplicaciones descentralizadas o voluntarias donde los datos no se entregan a un ente único. Se propone el uso de cifrado real en todas las comunicaciones, combinado con auditoría pública de los sistemas de inteligencia (para que no operen a discreción). Hay movimientos como *Data Trusts* o legislación de datos personales (GDPR en Europa) que buscan limitar el acceso irrestricto a datos, contrarrestando sistemas como Palantir.
- **Contra-vigilancia y autodefensa digital:** Para individuos preocupados por Palantir, las recomendaciones incluyen:
- Evitar redes sociales comerciales o minimizar la información pública, ya que Palantir explota dichas fuentes.
- Utilizar mensajería cifrada (como las apps seguras expuestas) para que los datos personales sensibles permanezcan inaccesibles.
- Vigilar a los legisladores y organismos de control; exigir transparencia sobre qué datos recolectan las agencias estatales.
- Apoyar iniciativas de datos descentralizados donde la persona controla qué comparte (por ejemplo, aplicaciones federadas, proyectos open data).
- Adoptar contramedidas técnicas: uso de wallets anónimos para transacciones, navegadores privados, sistemas operativos orientados a la privacidad (como Tails o Qubes OS para tareas especiales).
- **Resumen:** Palantir simboliza la capacidad técnica actual para la vigilancia omnipresente. Si bien no es un virus, sus plataformas demuestran cómo los datos masivos pueden usarse como arma contra la privacidad. La conclusión de este análisis es que la **soberanía y la autodefensa digital** deben contrarrestar no solo malware activo, sino también la apropiación pasiva de datos. Esto implica ejercer el derecho a la privacidad activamente, educar a la población y estructurar marcos legales robustos que limiten el uso de tecnologías intrusivas.

4.3 Metodologías de contra-vigilancia y autodefensa digital

Frente a las amenazas descritas (spyware como Pegasus, Big Data como Palantir, DPI, leyes invasoras), se requiere un enfoque de autodefensa electrónico:

- **Encriptación omnipresente:** Usar por defecto canales cifrados de extremo a extremo (aplicaciones seguras de la sección 1) para mensajería, y cifrar discos/almacenamiento de dispositivos para proteger datos locales. El cifrado en tránsito (HTTPS/TLS) y almacenamiento frena el acceso casual.
- **Sistemas operativos seguros:** Para comunicaciones sensibles, usar sistemas o modos de operación que reduzcan la huella de ataque:
- **Móviles**: activar bloqueos de arranque, evitar root/jailbreak, actualizar iOS/Android, considerar *GrapheneOS* o *CalyxOS* si es viable.
- **PCs**: sistemas orientados a privacidad (Tails, Qubes) para ciertas tareas, así aislando entornos de riesgo.
- **Anónimos y ofuscación:** Emplear Tor o VPNs confiables para ocultar el rastro de la dirección IP real. Sin embargo, hay que seleccionar VPN con política de no registros. Tor Browser para navegación y aplicaciones de mensajería (Session, Briar) que usan redes mix dificulta el rastreo.
- **Segmentación de dispositivos:** Separar la vida: usar dispositivos distintos para tareas críticas (trabajo sensible) y para uso cotidiano (redes sociales, juegos). Menos información en un solo dispositivo reduce el impacto si uno es atacado. Algunos sugieren tener un “teléfono burner” para contactos riesgosos, y no instalar apps sospechosas.
- **Monitoreo activo:** Revisar periódicamente el comportamiento del dispositivo: verificar permisos de apps, uso de batería inusual (indicador de spyware trabajando en segundo plano), tráfico de red extraño (monitorear con apps de análisis de datos). Aunque no es fácil detectar Pegasus, otros malwares simples sí, con apps de análisis de seguridad.
- **Actualización y patches:** Manter la infraestructura (hardware/firmware) actualizada. Las vulnerabilidades conocidas (CVE) son puertas de entrada para malware. Muchos ataques estatales aprovechan fallas recién descubiertas, pero se limitan cuando el SO está parchado.
- **Mínima exposición de datos:** Ser conscientes de la información compartida en redes sociales y servicios. Palantir y aliados se alimentan de datos públicos. Configurar privacidad en redes comerciales, o idealmente migrar a redes federadas abiertas (como las de sección 2) donde se exponga menos información personal.

- **Educación continua:** Estar al día con noticias de seguridad. Los ciberdelincuentes sofisticados (incluidos gobiernos) evolucionan rápido; la sociedad civil debe aprender a identificarlos. Organizaciones de seguridad (Electronic Frontier Foundation, Amnistía, etc.) suelen publicar guías para usuarios de a pie.
- **Comunidades y código abierto:** Favorecer la adopción de software libre auditable. El código abierto permite verificar que no hay puertas traseras incluidas. Participar o apoyar comunidades que desarrollan herramientas seguras asegura que estas evolucionen y sean auditadas.
- **Acción política y legal:** Aunque suena ajeno a "autodefensa digital" técnica, la presión para reformas legales es crucial. Apoyar iniciativas pro-transparencia, defender leyes de privacidad (como la ley 21.719) y rechazar normas que obliguen la instalación y el uso de softwares de espionaje y vigilancia.
- **Redundancia y resiliencia:** Tener múltiples canales de comunicación: si un medio falla (servicio bloqueado, app comprometida), tener alternativas listas (por ejemplo, cambiar rápido a XMPP si Telegram cae, o usar satélite de emergencia). Esta "pluri-herramienta" dificulta que un actor supremo silencie todos los caminos.
- **Conclusión práctica:** La autodefensa digital exige una postura proactiva. No basta con usar apps seguras; es necesario un **hábito de seguridad** integral: desde las contraseñas robustas, hasta la elección de hardware confiable. El objetivo es crear capas de protección, de modo que si un elemento falla, no todo el sistema colapse. La conciencia de que organismos estatales disponen de recursos enormes debe motivar a la ciudadanía organizada (via comunidades y ONG tecnológicas) a seguir investigando y compartiendo contramedidas. Solo con una cultura de seguridad colectiva se puede mejorar significativamente la resistencia electrónica.

Conclusiones

La transición hacia una comunicación móvil soberana y segura es un proceso complejo que requiere conocimiento, herramientas y vigilancia permanente. En este informe se ha revisado exhaustivamente un amplio conjunto de aplicaciones y tecnologías, así como las amenazas y contramedidas más relevantes en el panorama actual.

Las aplicaciones de mensajería instantánea libres, cifradas y descentralizadas ofrecen un modelo de defensa heterogéneo contra los diversos aparatos de vigilancia más agresivos que se están desarrollando e implementando en Chile.

Cada una de estas aplicaciones presenta un modelo de cifrado, enrutamiento y protocolos distintos. Mientras algunas destacan por su comodidad y facilidad de uso manteniendo fuertes modelos de seguridad con robustos algoritmos de cifrado, otras alternativas ofrecen propuestas mucho más extremas y radicales que sacrifican un poco de comodidad para ofrecer mayor seguridad. No hay "solución única": la elección depende de la amenaza particular que se busque mitigar. Lo crucial es que el usuario comprende los trade-offs: algunas aplicaciones sacrifican inmediatez o facilidad por mayor privacidad, otras privilegian la experiencia usuario sobre controles avanzados.

En el ámbito de las redes sociales, la tendencia hacia plataformas federadas y descentralizadas representa un contraste con las plataformas comerciales. Éstas empoderan al individuo al permitirle poseer sus datos y participar en comunidades con reglas definidas localmente. También fomentan la pluralidad de discurso, al erosionar el monopolio de contenido de unos pocos gigantes. Sin embargo, implican responsabilidades: cada instancia o comunidad puede tener normas diferentes, y la calidad de experiencia depende de la infraestructura de cada servidor. Lo más valioso es que estas redes disipan puntos únicos de fallo: en caso de censura o censura de un servidor, los usuarios pueden cambiarse a otro. Esto solidifica la resistencia social frente a los intentos de silenciar voces independientes.

El análisis de las políticas y aparatos de vigilancia estatales revela un panorama preocupante. Las leyes que amplían las capacidades de inteligencia, tecnologías de vigilancia y espionaje masivo demuestran que los derechos a la integridad, confidencialidad y autenticidad están siendo violadas. La legislación puede otorgar permisos extensos para inspeccionar comunicaciones, a veces sin controles eficientes. Esto socava la privacidad personal y la libre expresión. Sin embargo, la presión pública, la visibilidad de casos documentados y la contrainsurgencia digital (uso de criptografía y redes descentralizadas) pueden actuar como freno a esta tendencia. Es indispensable demandar mecanismos de transparencia y rendición de cuentas en la vigilancia gubernamental.

En el ámbito del malware y la vigilancia avanzada, amenazas como Pegasus demuestran que ningún dispositivo es completamente seguro si un adversario suficientemente equipado decide atacarlo. La infección de teléfonos móviles ha significado que la seguridad en el extremo emisor se está volviendo tan crítica como el cifrado de datos en tránsito. El uso de herramientas analizadas aquí (por ejemplo, prefiriendo aplicaciones con cifrado E2E, manteniendo dispositivos actualizados, adoptando sistemas operativos reforzados) es la primera línea de

defensa. Por otro lado, sistemas de big data como Palantir plantean retos aún más abstractos: no podemos cifrar la atmósfera informativa. En respuesta, la sociedad debe recopilar propuestas para gobernar el uso de la inteligencia artificial y big data en la esfera pública, equilibrando seguridad con derechos individuales.

Recomendaciones finales: Los usuarios y organizaciones deben integrar a la mayor brevedad prácticas de autodefensa digital: migrar a servicios de mensajería seguros (especialmente para temas sensibles), aprovechar redes federadas en lugar de centrales, y mantenerse informados sobre actualizaciones de seguridad. Las comunidades tecnológicas deben continuar desarrollando herramientas más amigables que unan seguridad con usabilidad. Paralelamente, la presión ciudadana para reformas legislativas y la exigencia de auditorías independientes sobre la vigilancia gubernamental son también parte clave de la resistencia. En definitiva, la soberanía y autodefensa electrónica no dependen solo de la tecnología: requieren una convergencia de cultura digital, responsabilidad institucional y vigilancia pública.

Con este informe, producto del Taller de Privacidad en Teléfonos Móviles, se espera que el lector comprenda en profundidad tanto las soluciones disponibles para proteger la información en los teléfonos móviles, como los riesgos que afrontamos frente a actores poderosos. En la intersección de innovación tecnológica y derechos civiles, está la clave para avanzar hacia un futuro donde la comunicación permanezca libre y privada, incluso bajo la presión de la vigilancia constante.